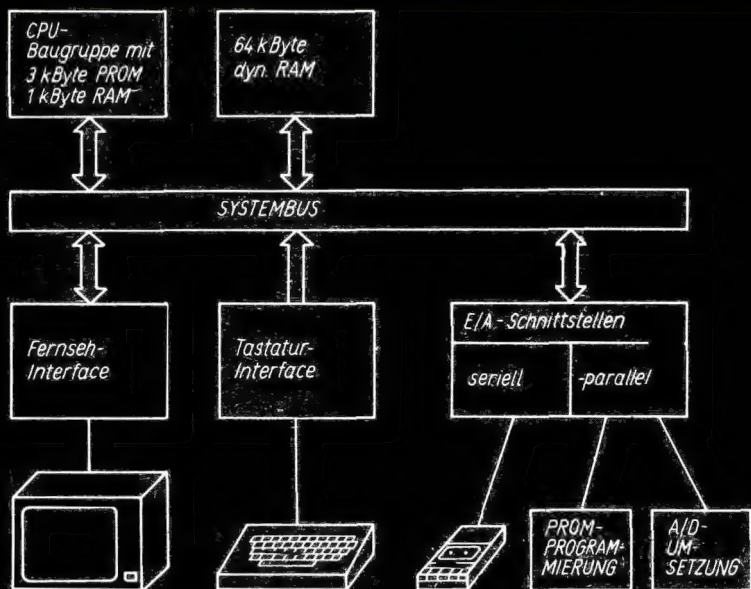


amateurreihe

electronica



**Bernd Hübler
Klaus-Peter Evert**

**Ausbaufähiger
Mikrocomputer
mit dem U 880**

**227/
228**

electronica · Band 227/228

BERND HÜBLER · KLAUS-PETER EVERT †

Ausbaufähiger Mikrocomputer mit dem U 880

MILITÄRVERLAG
DER DEUTSCHEN DEMOKRATISCHEN
REPUBLIK

Bernd Hübler/Klaus-Peter Evert:
Ausbaufähiger Mikrocomputer mit dem U 880.
– 1. Auflage – Berlin: Militärverlag der DDR, 1985 –
(electronica 227/228)

1. Auflage, 1985
© Militärverlag der Deutschen Demokratischen Republik (VEB) – Berlin, 1985
Lizenz-Nr. 5
Printed in the German Democratic Republic
Gesamtherstellung: Druckerei Märkische Volksstimme Potsdam
Lektor: Rainer Erlekamp
Zeichnungen: Angelika Ulsamer
Typografie: Martina Schwarz
Redaktionsschluß: 12. November 1984
LSV 3539
Bestellnummer: 746 728 1
00380

Inhaltsverzeichnis

1.	Vorwort	5
2.	Systembeschreibung	7
3.	CPU-Baugruppe	10
3.1.	Taktversorgung, Buspufferung	10
3.2.	Speicher	14
3.3.	Zentrale Funktionseinheiten	15
3.3.1.	Speicherselektierung	15
3.3.2.	I/O-Portselektierung	17
3.3.3.	Startlogik	18
3.4.	Aufbau und Inbetriebnahme	22
4.	Speichererweiterung	25
4.1.	ROM-Speicher	26
4.2.	RAM-Speicher	30
4.2.1.	Statischer RAM-Speicher	30
4.2.2.	Dynamischer RAM-Speicher	32
4.3.	Aufbau und Inbetriebnahme	38
5.	Fernsehinterface	42
5.1.	Zeichendarstellung auf dem Bildschirm	42
5.2.	Schaltungsbeschreibung	46
5.2.1.	Bildwiederholtspeicher, Adreßmultiplexer	48
5.2.2.	Synchron- und Austastsignalerzeugung, Adressen- bereitstellung	52
5.2.3.	Videosignalerzeugung	55
5.2.4.	BAS-Signalerzeugung	61
5.3.	Erweiterungsmöglichkeiten	61
5.3.1.	Zeichengenerator mit Graphiksymbolen	62
5.3.2.	RAM-Speicher als Zeichengenerator	62
5.3.3.	Vollgraphik	62
5.4.	Aufbau und Inbetriebnahme	63
6.	Alphanumerische Tastatur	65

6.1.	Schaltungsbeschreibung	65
6.2.	Aufbau und Inbetriebnahme	70
7.	Ein-/und Ausgabebaugruppen	74
7.1.	Universelle I/O-Karte	74
7.2.	EPROM-Programmiergerät	79
7.2.1.	Programmiervorschriften	79
7.2.2.	Die Schaltung	81
7.2.3.	Aufbau und Codierung	83
7.2.4.	Ein Beispielprogramm	85
7.3.	Analogschnittstellen	87
7.3.1.	Digital/Analog-Wandler	87
7.3.2.	Analog/Digital-Wandler	91
8.	Kassetteninterface	95
8.1.	Aufzeichnungsverfahren	95
8.2.	Realisierung	97
9.	Monitorprogramm	104
9.1.	Beschreibung der Monitorkommandos	104
9.2.	Schnittstellen und Erweiterbarkeit	110
9.3.	Das Programm	113
10.	Hardwareaufbau und Inbetriebnahme	177
10.1.	Stromversorgung	177
10.2.	Systemaufbau	179
11.	Anhang	184
12.	Literaturverzeichnis	188

1. Vorwort

Die Möglichkeiten und Erfordernisse der modernen Mikroelektronik zeigen sich im Bereich der Mikrocomputertechnik besonders deutlich.

Der Anwendungsbereich des Mikrocomputers reicht von der Steuerung und Regelung industrieller Prozesse – genannt sei hier nur das Schlagwort Industrieroboter – über den wissenschaftlich-technischen und den Bürocomputer bis hin zum Heimcomputer. Immer stärker dringt der Mikrocomputer auf Grund seiner ständig wachsenden Leistungsfähigkeit in die Bereiche der traditionellen EDV ein.

Für ein modernes Industrieland ist es unumgänglich, die Möglichkeiten dieser neuen Technik zu nutzen.

Wissen über Computertechnik im allgemeinen und Mikrocomputertechnik im speziellen sowie computerspezifische Denkweisen dürfen deshalb nicht einem kleinen Kreis von Spezialisten vorbehalten bleiben. Diese Erkenntnis hat international zu einer bemerkenswerten Verbreitung und Popularisierung des Heimcomputers bzw. des Personalcomputers besonders bei jugendlichen Interessenten geführt.

Das vorliegende Heft wendet sich deshalb an den elektronisch versierten Amateur, aber auch an den Elektroniker, der noch keine Gelegenheit hatte, auf dem Gebiet der Mikrocomputertechnik praktische Erfahrungen zu sammeln. Grundkenntnisse der Digitaltechnik sowie der Mikroprozessortechnik, wie sie beispielsweise [1] vermittelt, werden vorausgesetzt.

Das gegenwärtig in der DDR verfügbare Schaltkreisangebot gestattet dem potentiellen Interessenten den Aufbau eines leistungsfähigen Mikrocomputers. Nachbaufähige Schaltungen für die Realisierung der dazu benötigten Hardware sowie erprobte Betriebssoftware sollen den Weg zum fertigen Gerät erleichtern.

Der Aufbau des Systems ist mit einiger Mühe verbunden und erfordert ein gewisses Durchstehvermögen.

Beim eigenen Realisieren begreift man jedoch die relativ komplizierte Hardware am besten. Gerade für den Bereich des Mikro-

computers sind gründliche Hardwarekenntnisse von besonderem Wert.

Die Autoren hoffen, daß einem möglichst breiten interessierten Personenkreis der Einstieg in die zukunftssträchtige Problematik der Mikrocomputertechnik ermöglicht wird.

Berlin, im Januar 1984

Bernd Hübler
Klaus-Peter Evert †

2. Systembeschreibung

Das in der Beschreibung vorgestellte Mikrocomputerkonzept basiert auf der Schaltkreisfamilie des Mikroprozessors *U 880*. Diese leistungsfähige 8-Bit-CPU einschließlich der dazugehörigen Peripherieschaltkreise sowie der bereits verfügbaren Speicher erlaubt bei vertretbarem Aufwand den Aufbau eines universell einsetzbaren Mikrocomputers.

Bild 2.1 zeigt den Übersichtsschaltplan der Hardware des Systems. Danach besteht das voll ausgebaute System aus folgenden Komponenten:

- CPU-Baugruppe (realisiert als Einplatinenrechner) mit 3-kByte-PROM, 1-kByte-RAM
- 64-kByte-RAM-Speicher
- Bildschirm mit 24 Zeichenzeilen zu je 64 Zeichenpositionen
- alphanumerische Tastatur (maximal 64 Tasten)
- Kassette als externes Speichermedium
- parallele und serielle Ein-/Ausgabeschnittstelle
- PROM-Programmiergerät

Als Bildschirm eignet sich ein handelsüblicher Schwarzweißfernsehempfänger (möglichst Koffergerät), und zur Bandaufzeichnung nutzt man einen Kassettenrecorder.

Die Hardware kann schrittweise, beginnend mit Aufbau und Inbetriebnahme des Einplatinenrechners bis hin zum kompletten System, ausgebaut werden.

Das Systemkonzept des beschriebenen Mikrocomputers soll einen möglichst universellen Anwendungsbereich sichern. Dementsprechend kann man ein breites Softwarespektrum einsetzen. Es reicht vom Maschinensprachniveau bis zu Assemblerprogrammen und zu höheren Programmiersprachen (z. B. BASIC). Aus diesem Grund steht der RAM-Speicher ab Adresse 0 zur Verfügung. Ein möglichst weitgehender Ausbau des RAM-Bereichs erweist sich unter den genannten Aspekten als günstig.

Der Computer soll aber auch als Entwicklungssystem genutzt werden. Für diesen Anwendungsfall wird ein Gerät vorgestellt, das die Programmierung aller wichtigen, derzeit bekannten PROM-Typen (≥ 1 kByte) gestattet.

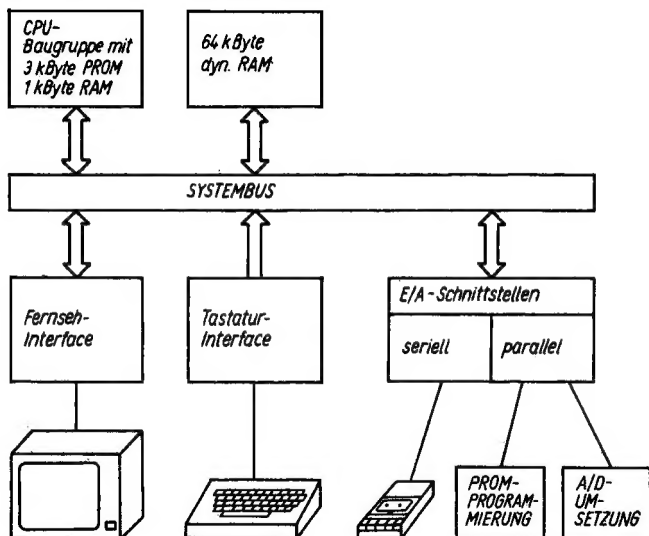


Bild 2.1 Hardwareübersicht

Voraussetzung für die Arbeit mit dem Computer ist das Vorhandensein eines (im allgemeinen) systemresidenten Betriebssystems. Als Minimalforderung muß ein Urlader auf dem System abgespeichert sein, der das Laden eines Betriebssystems von einem externen Speichermedium ermöglicht. Für den vorgestellten Mikrocomputer wurde ein den Belangen des Systems angepaßtes Monitorprogramm entwickelt. Mit Hilfe des Monitors wird der Datenverkehr zwischen CPU und Peripherie abgewickelt und das Arbeiten auf Maschinensprachniveau ermöglicht. Es lassen sich also Programme sowie Datenein- und -ausgaben testen. Darüber hinaus verfügt das Monitorprogramm über Schnittstellen, die es gestatten, Teile desselben in Anwenderprogrammen mitzunutzen. Beispielsweise können die auch beim Betrieb mit höheren Programmiersprachen benötigten Treiberrouninen (Bildschirm, Tastatur, Kassettenaufzeichnung) so in einfacher Weise angesprochen werden.

Unter Berücksichtigung des Systemkonzepts sollte der Monitor das einzige systemresidente, d. h. auf PROM abgespeicherte, Pro-

gramm sein. Alle weitere Software kann vom externen Speichermedium (Kassette) bei Bedarf in den Arbeitsspeicher des Mikrocomputers geladen werden. Abhängig von den Möglichkeiten (z. B. hinsichtlich des RAM-Speicherausbaus) lassen sich jedoch auch individuelle Lösungen realisieren. Beispielsweise kann ein geeigneter BASIC-Interpreter auch auf PROM abgespeichert werden. Die schaltungsmäßigen Lösungen für den in diesem Fall zusätzlichen Bedarf an PROM-Speicherkapazität sind im vorgestellten Hardwareumfang enthalten.

Die frei verfügbaren E/A-Schnittstellen gestatten das Koppeln des Mikrocomputers mit externen Geräten. Für die in diesem Zusammenhang häufig benötigten A/D-Wandler bzw. D/A-Wandler werden Lösungsvorschläge unterbreitet.

Im voll ausgebauten Zustand steht damit ein leistungsfähiger Mikrocomputer zur Verfügung, der darüber hinaus hard- und softwaremäßig für individuelle Erweiterungen vorbereitet ist.

3. CPU-Baugruppe

Die CPU-Baugruppe stellt, wie bereits der Name sagt (*Central-Processing-Unit* – zentrale Verarbeitungseinheit), das Herzstück des Mikrocomputersystems dar. Basierend auf dem Mikroprozessor *U 880*, bestimmt sie in wesentlichem Umfang die Leistungsfähigkeit des Systems.

Als Voraussetzung zur Erfüllung ihrer Hauptfunktion hat die CPU-Baugruppe die Aufgabe der Taktversorgung des Mikroprozessors sowie der Verteilung der Systeminformationen (Daten, Adressen, Steuersignale) im Computer. Ihre spezielle Konfiguration ist vorrangig vom gewünschten Einsatzzweck (z. B. Heimcomputer, komfortabler Bürocomputer, Steuercomputer usw.) abhängig und damit auch vom Umfang und Niveau der Software, die man beabsichtigt, auf dem Computer lauffähig zu machen (fachmännisch ausgedrückt: zu implementieren). Letztlich ist die CPU-Platine auch eine Widerspiegelung der Systemphilosophie des Entwicklers.

Bild 3.1 zeigt die für diese Broschüre gewählte Variante.

3.1 Taktversorgung, Buspufferung

Als Taktgenerator wird ein quarzstabilisierter TTL-Generator verwendet. Er schwingt mit 10 MHz und liefert der CPU, nachdem diese Frequenz 1 : 4 untersetzt und durch 2 Leistungsgatter *D 240* gepuffert wurde, den Standardtakt von 2,5 MHz. Außerdem wird der Takt am Systembus bereitgestellt.

Wer über einen geeigneten Schwingquarz verfügt, hat die Möglichkeit, mit der angegebenen Schaltung oder entsprechend anders eine Taktfrequenz von genau 2,4576 MHz zu erzeugen. Damit lassen sich standardisierte Datenübertragungsraten (Baudraten) für die entsprechende Peripherie (Kassettenaufzeichnung, Datenfernübertragung usw.) realisieren.

Die an den RESET-Eingang des Mikroprozessors angeschlossene Kombination aus RC-Glied und den Gattern *D3.1* und *D3.2* bewirkt das Einschalt-RESET (Power-On-RESET) und damit den

Urstart des Systems. Diese RESET-Schaltung wird häufig auch mit einfachen NAND-Gattern bzw. ohne Gatter ausgeführt. Die Autoren empfehlen die Verwendung des Triggerschaltkreises *MH 7413*. Auch das RESET-Signal steht am Systembus zur Verfügung. Zum Realisieren einer Reset-Taste kann man eine Schaltungslösung verwenden, die einen Datenverlust bei eventuell eingesetzten dynamischen RAM (siehe Abschnitt 4.2.2.) verhindert. Eine mögliche Variante ist [2] zu entnehmen. Die restlichen CPU-Eingänge INT, NMI, BUSRQ, WAIT – ihre Bedeutung wird als bekannt vorausgesetzt und ist z. B. in [1], [2] nachzulesen – werden, da sie L-aktiv sind, für den Fall des nicht angesteuerten Zustandes mit den entsprechenden Widerständen auf ein sicheres H-Potential gebracht.

Die Daten- und Adreßausgänge der CPU *U880* sind als Tri-State-Ausgänge ausgeführt. Sie sind ebenso wie die L-aktiven Steuerausgänge – von denen nur die Signale RD, WR, MREQ, IORQ über Tri-State-Ausgänge verfügen – in der Lage, eine TTL-Lasteinheit zu treiben. Für das Vergrößern der Busbelastbarkeit aus der Sicht der CPU sowie zur Leistungstreibung, d. h. zum Unterdrücken des Einflusses von Störimpulsen und parasitären Kapazitäten, muß man die genannten Signale vor dem Bereitstellen auf den geeigneten Treiberschaltkreis puffern.

Der Datenverkehr zwischen CPU und Peripherie ist grundsätzlich bidirektional, d. h., er erfolgt in beiden Richtungen.

Dementsprechend werden die Datenausgänge $D0 \dots D7$ mit 2 bidirektionalen Treiberschaltkreisen des Typs *8216* (*D11*, *D12*) gepuffert. Dazu ist es notwendig, gleichwertige Datenein- und -ausgänge (*DI* und *DO*) des *8216* parallelzuschalten. Für das Durchschalten in der jeweils gewünschten Richtung, d. h. für das Bereitstellen des Datenrichtungssignals DIEN am Schaltkreis *8216*, ist die Datenrichtungslogik, bestehend aus dem Gatter *D4.3*, zuständig. Wenn das Lesesignal RD aktiv ist, ist also der Datenfluß in der Richtung Peripherie \rightarrow CPU möglich. Für den Fall $\overline{RD} = H$ sind die Verhältnisse umgekehrt.

Das M1-Signal muß bei der Datenrichtungsumschaltung berücksichtigt werden, damit der Interruptmode 2 (Vektorinterrupt) abgesichert ist. Dabei muß die CPU einen Interruptvektor von der Peripherie lesen. In diesem Fall ist RD nicht aktiv, sondern M1 und IORQ.

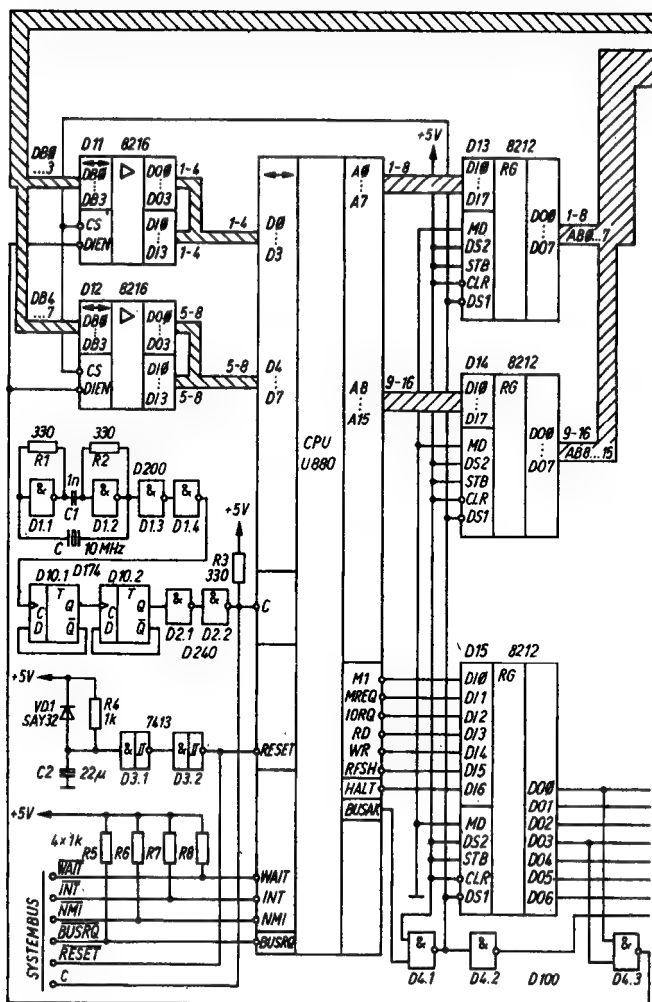


Bild 3.1 CPU-Baugruppe

Als Treiber für die Adreßausgänge A0 ... A15 der CPU setzt man Schaltkreise des Typs 8212 (D13, D14) ein.

Der Schaltkreis 8212 besteht im wesentlichen aus einem 8-Bit-Datenregister und -treiber mit Zusatzlogik sowie Tri-State-Ausgangsstufen. Er ist auf Grund seiner Konzeption (löst auch über ein internes Flip-Flop Interrupt aus) universell zur Bussteuerung und Datenpufferung geeignet.

Im vorliegenden Einsatz ist der 8212 eigentlich unterfordert. Er wurde aus Gründen der guten Verfügbarkeit trotzdem eingesetzt. Die Steuersignale $\overline{M1}$, \overline{RD} , \overline{WR} , \overline{MREQ} , \overline{IORQ} , \overline{HALT} sowie das Refresh-Signal, das man zum „Auffrischen“ der dynamischen RAM benötigt, werden über einen weiteren Schaltkreis des Typs 8212 gepuffert.

Mit dem Signal $\overline{BUSA\overline{K}} = L$ quittiert die CPU die Anmeldung einer DMA-Anforderung am Eingang \overline{BUSRQ} . Für diesen Fall sind die Daten- und Adreßausgänge sowie die Steuersignalausgänge \overline{RD} , \overline{WR} , \overline{MREQ} , \overline{IORQ} hochohmig und die weiteren Ausgänge inaktiv. Beim DMA-Betrieb, d. h. beim direkten Speicherzugriff (*Direct Memory Access*) einer peripheren Einheit, ist die CPU bekanntlich unbeteiligt. Deshalb muß sie vom Bus abgeschaltet werden. In der Schaltung gemäß Bild 3.1 wird der aktive Zustand von $\overline{BUSA\overline{K}}$ am Ausgang von Gatter D 4.1 zum Deselektieren der Bustreiber 8216 und 8212 benutzt. Damit sind alle Daten-, Adreß- und Steuerleitungen (außer $\overline{BUSA\overline{K}}$) hochohmig vom Bus getrennt.

3.2. Speicher

Bild 3.1 zeigt, daß die vorgestellte CPU-Baugruppe mit 3-kByte-Programmspeicher und 1-kByte-Arbeitsspeicher ausgerüstet ist. Zusammen mit den bereits beschriebenen Schaltungselementen steht damit also ein in sich funktionsfähiger Einplatinencomputer zur Verfügung.

Der Anschluß der verwendeten EPROM-Schaltkreise U 555 (2708) sowie RAM-Schaltkreise U 202 (2102) an den Daten- und Adreßbus geht ebenfalls aus Bild 3.1 hervor. Man stellt fest, daß eine Zusammenstellung gleichwertiger Datenein- und -ausgänge (DI, DO) beim RAM U 202 nicht möglich ist. Aus diesem Grund werden die Datenausgänge DO nur für den Fall, daß die CPU ei-

nen Speicherlesezyklus zum selektierten RAM-Block durchführt, über die Tri-State-Ausgänge des Schaltkreises 8212 (D15) auf den Datenbus geschaltet. Andernfalls sind sie hochohmig vom Bus getrennt. Günstig, besonders wenn man größere Speicherblöcke auf der Basis statischer RAM des Typs *U 202* verwendet, ist das Puffern sowohl der Dateneingänge als auch der Datenausgänge. Hierfür bietet sich wieder der bidirektionale Treiberschaltkreis 8216 an. Die Verdrahtung der benötigten Schaltkreise 8216 mit den Ein- bzw. Ausgängen des *U 202* ergibt sich aus den jeweiligen Pin-Belegungen (siehe auch Bild 4.3).

Die Lese/Schreib-Umschaltung am \overline{WE} -Eingang des *U 202* sowie die Bereitstellung des für die gewünschte Funktion des 8212 notwendigen Signals DS2 realisiert man mit geringem Aufwand über die NOR-Gatter D 5.1, D 5.2, D 5.3. Dazu werden die CPU-Signale \overline{RD} , \overline{WR} verwendet.

Beim Einsatz des 8216 steht als Datenrichtungssignal \overline{DIEN} das \overline{RD} -Signal zur Verfügung (siehe auch Bild 4.3).

Für die Chipselektierung der 3 EPROM *U 555* sowie des 1-kByte-RAM-Blocks ist eine 1-aus-4-Decodierung notwendig. Sehr einfach und übersichtlich läßt sich dies mit dem 1-aus-10-Decoder *MH 7442* lösen, der das jeweilige Chipselektsignal aus den Adressen AB10, AB11 sowie einem 4-kByte-Blockselektsignal decodiert. Aber auch jeder andere 1-aus-n-Decoder ($n \geq 4$) ist geeignet (z. B. 1-aus-8-Decoder 8205).

Das CS-Signal für die eventuell eingesetzten bidirektionalen Treiber 8216 ist mit dem für den 1-kByte-RAM-Block identisch.

3.3. Zentrale Funktionseinheiten

Die 3 auf der CPU-Platine untergebrachten Funktionseinheiten haben in engerem Sinne mit der Funktion der CPU-Baugruppe nichts zu tun, sie sind aber für das Systemkonzept von Bedeutung.

3.3.1. Speicherselektierung

Die Speicher werden bei üblichen Mikrocomputersystemen (z. B. dem *K 1520* vom VEB Kombinat *Robotron*) häufig dezentral se-

lektiert, d. h., mit entsprechenden Decodierschaltungen erzeugt sich jede Speicherplatine ihren Adreßbereich im Adreßraum des Computers (64 kByte beim *U 880*) gewissermaßen selbst. Dieser Adreßbereich ist meistens vom Anwender über Drahtbrücken oder DIL-Schalter auswählbar.

Im Gegensatz dazu wurde für das vorgestellte System eine Schaltungsvariante zur zentralen Speicherselektierung entwickelt. Als elegante Lösung bietet sich der 1-aus-16-Decoderschaltkreis *MH 74154* an, der es erlaubt, aus den Adressen AB12 ... AB15 16 4-kByte-Blockselektsignale zu decodieren. Damit ist man in der Lage, den 64-kByte-Adreßbereich der CPU *U 880* in 16 4-kByte-Speicherblöcke aufzuteilen. Tabelle 3.1. verdeutlicht dies.

Für die praktische Auswahl eines gewünschten Speicherbereichs ist es lediglich notwendig, den jeweiligen Ausgang des *MH 74154* mit dem Blockselekteingang der entsprechenden Speicherplatine galvanisch zu verbinden.

Die 4-kByte-Blöcke werden dann dezentral mit Hilfe der angegebenen Chipselektsschaltungen in die gewünschte Größe unterteilt. In der Schaltung gemäß Bild 3.1 wird der Eingang G1 des *MH 74154* als notwendige logische Verknüpfung der Speicherselektierung mit dem $\overline{\text{MREQ}}$ -Signal der CPU verwendet. Den Eingang G2 nutzt man zum Deselektieren des gesamten Speicherbereichs ($\text{G2} = \text{H}$). Der Grund für diese Maßnahme wird im Zusammenhang mit der Startlogik erläutert.

Tabelle 3.1. Zentrale Speicherselektierung

Ausgang <i>MH 74154</i>	decodierter Speicherbereich (HEXA)	selektierter Speicherblock (kByte)
0	0000–0FFF	0–4
1	1000–1FFF	4–8
2	2000–2FFF	8–12
.	.	.
.	.	.
.	.	.
15	F000–FFFF	60–64

3.3.2. I/O-Portselektierung

Der Datenverkehr der CPU mit peripheren Geräten wird, sofern diese nicht wie gewöhnliche Speicherbereiche ansprechbar sind, über I/O-Ports (Ein-/Ausgabe-Tore) abgewickelt. Die Adressierung der I/O-Ports ergibt sich aus dem speziellen Verhalten der CPU *U880*. Diese liefert während eines I/O-Zyklus ($\overline{\text{IORQ}} = \text{L}$) die 8 niederwertigen Adreßbit A0 ... A7 als I/O-Portadresse. Ob es sich um einen Eingabe- (Portlesen) oder Ausgabezyklus (Portschreiben) handelt, richtet sich nach der gleichzeitigen Aktivität des entsprechenden CPU-Signals $\overline{\text{RD}}$ oder $\overline{\text{WR}}$ (siehe [1], [2]).

Mit den 8 zur Verfügung stehenden Adreßbit A0 ... A7 können also maximal 256 I/O-Ports durch die CPU angesprochen werden. Die I/O-Ports werden im vorliegenden System folgendermaßen selektiert:

Als Decoderschaltkreis verwendet man wieder den von der Speichersелеktierung bekannten 1-aus-16-Decoder *MH 74154*. Die Adreßbit AB0, AB1 werden im allgemeinen für spezielle Selektierungsfunktionen im Zusammenhang mit den I/O-Schaltkreisen *PIO U855* bzw. *SIO U856* benötigt (siehe Abschnitte 7. und 8.) und sind damit bereits belegt. Aus den in der Wertigkeit nächstfolgenden Adressen AB2, AB3, AB4, AB5 wird das I/O-Portselektsignal mit dem *MH 74154*, wie in Tabelle 3.2. angegeben, decodiert. Analog zur Speichersелеktierung wird dabei die notwendige logische Verknüpfung mit dem Signal $\overline{\text{IORQ}}$ über den Toreingang G1 erreicht.

Damit verfügt man über 16 ($\times 4$) I/O-Portadressen. Die Unterscheidung der an jedem Ausgang vorliegenden 4 Einzeladressen ist

Tabelle 3.2. I/O-Portselektierung

Ausgang <i>MH 74154</i>	decodierte I/O-Ports (HEXA)
0	00, 01, 02, 03
1	04, 05, 06, 07
2	08, 09, 0A, 0B
.	.
.	.
.	.
15	3C, 3D, 3E, 3F

mit zusätzlichem Decodieraufwand (jeweils 1 aus 4) möglich. Für das System sind 16 I/O-Ports ausreichend, zumal bei Verwendung der systemspezifischen I/O-Schaltkreise PIO *U855* bzw. SIO *U856* über jedes der 16 Ports weitere schaltkreisinterne I/O-Kanäle angesprochen werden können (Port B/ \bar{A} Select über CPU-Adresse A0).

Ein gewünschtes I/O-Port wird der jeweiligen peripheren Einheit wieder durch Verbinden des entsprechenden Ausgangs des *MH 74154* mit dem Chipselekteingang der peripheriespezifischen Interfacebaugruppe (z. B. Chip-Enable-Eingang beim PIO- oder SIO-Schaltkreis) zugeordnet.

3.3.3. Startlogik

Der Mikroprozessor *U880* kann grundsätzlich über $\overline{\text{RESET}}$ oder über Interrupt gestartet werden (siehe [1]). Für den Urstart des Systems verwendet man üblicherweise das bereits erwähnte „Power-On-RESET“. Hierbei beginnt die CPU die Befehlsbearbeitung bei der Speicherzelle 0. Aus diesem Grund ist es häufig so, daß der Beginn des Betriebssystems des Mikrocomputers (z. B. eines Monitorprogramms) auf die Speicheradresse 0 gelegt wird. Diese Methode ist mit einigen Nachteilen verbunden, die den Ausschlag für die Wahl einer anderen Lösung gaben. Die genannten Nachteile sind vor allem softwarebedingt und ergeben sich daraus, daß im oben beschriebenen Fall (d. h. Monitorbeginn bei Adresse 0) ab Adresse 0 kein frei verfügbarer Speicherbereich vorhanden ist. Dies soll im folgenden kurz erläutert werden.

Für Mikrocomputer auf der Basis des *U880* ist heute ein breites Softwareangebot bekannt, das von Assembler- und Disassemblerprogrammen bis zu Interpreter- bzw. Compilerprogrammen für höhere Programmiersprachen (z. B. BASIC, PASCAL, FORTRAN usw.) reicht. In vielen Fällen läuft diese Standardsoftware ab Adresse 0. Ein Umschreiben bewährter Programme für andere Speicherbereiche ist mit großem Aufwand und entsprechend vielen Fehlermöglichkeiten verbunden. Als weiterer Aspekt ist zu beachten, daß einige der genannten Programme nur auf RAM-Speichern lauffähig sind, da sich in diesem Fall die notwendigen Arbeitszellen innerhalb des jeweiligen Programmspeicherbereichs befinden (also nicht in andere RAM-Bereiche ausgelagert sind).

Im Anfangsbereich des Adreßraums des *U880* befinden sich einige Adressen mit besonderer Bedeutung (*00H*, *08H*, *10H* ... *38H*, *66H*: Restart bzw. Interruptmode 1 und *NMI*), die möglichst allgemein verfügbar sein sollten.

Aus den genannten Gründen wurde im System die Voraussetzung dafür geschaffen, daß – beginnend bei der Adresse *0* – ein möglichst großer, zusammenhängender RAM-Bereich realisiert werden kann. In diesem Zusammenhang müssen noch ein paar Überlegungen zur Speicherbereichsaufteilung angestellt werden:

Man benötigt für das System unbedingt ein Monitorprogramm (siehe Abschnitt 9.), welches das Arbeiten auf Maschinensprachniveau sowie die Handhabung aller weiteren Software ermöglicht. Als wichtiges, den Dialog zwischen Mensch und Computer unterstützendes peripheres Gerät wird ein Fernsehinterface benötigt (siehe Abschnitt 5.). Wie sich zeigen wird, kann man dieses durch den Computer wie einen normalen RAM-Speicherbereich ansprechen.

Im Ergebnis der Überlegungen kommt man zu der im Bild 3.2 angegebenen Speicherbereichsaufteilung.

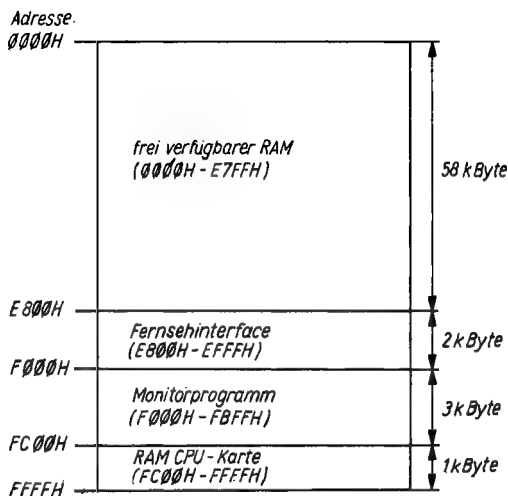


Bild 3.2 Speicherbereichsaufteilung

Für den Bildwiederholpeicher des Fernsehinterface werden 2-kByte im Adreßbereich E800H . . . EFFFH reserviert. Das in Abschnitt 9. beschriebene Monitorprogramm wird im Bereich F000H FC00H untergebracht. In den bis zum Ende des Adreßraums verbleibenden 1-kByte-Bereich paßt genau der Arbeitsspeicher (1-kByte-RAM) der CPU-Platine.

Damit sind die oberen 6 kByte des 64-kByte-Adreßbereichs des Computers belegt, und es steht ein zusammenhängender Adreßraum von 58 kByte (0000H . . . E7FFH) für Anwenderprogramme zur freien Verfügung.

Bei der gewählten Speicheraufteilung muß der Beginn der Programmbearbeitung bei der vereinbarten Startadresse des Monitorprogramms F000H gewährleistet werden. Hierfür gibt es hard- und softwareorientierte Möglichkeiten. Beispielsweise kann man den Monitor einschließlich einer einfachen Laderoutine ab Adresse 0 auf PROM unterbringen. Die Laderoutine dupliziert dann nach Netzeinschalten den Monitor zur Adresse F000H, wo in diesem Fall natürlich ein RAM-Speicher zur Verfügung stehen muß. Nach Beendigung dieses Vorgangs wird der PROM deselektiert, damit ab Adresse 0 ein frei verfügbarer Speicher vorhanden ist. Auf weitere Einzelheiten dieser Variante soll nicht eingegangen werden, da in dieser Broschüre die in Bild 3.1 angegebene einfache Hardwarelösung verwendet wird.

Die Funktionsweise der Startlogik soll an Hand des Impulsbildes (Bild 3.3) erläutert werden.

Mit dem Power-On-RESET wird das RS-Flip-Flop (D 6.2, D 6.3) gesetzt, alle Datenleitungen werden über Open-Collektor-Schaltkreise D 7, D 8 auf L gezogen. Das bedeutet, daß die CPU – beginnend bei der Adresse 0 – mit jedem Befehlsholezyklus (M1-Zyklus) den Operationscode 00H, d. h. den Befehl NOP, einliest. Mit NOP führt die CPU bekanntlich einen Leerzyklus aus. In dieser Weise wird der Adreßbereich 0000H . . . EFFFH durchlaufen. Bei Erreichen der Adresse F000H (unter logischer Verknüpfung mit MREQ) erscheint am Ausgang von Gatter D 9 L-Potential, das RS-Flip-Flop kippt zurück, und der Datenbus wird für normale Befehlsbearbeitung freigegeben.

Um einen Kurzschluß von Speicherausgängen beim Hochstartvorgang zu vermeiden (DB0 . . . DB7 = L!), benutzt man das Signal MEMDES am Ausgang von Gatter D 2.4 dazu, über den Toreingang G 2 der zentralen Speichersелеktierung sämtliche angeschlos-

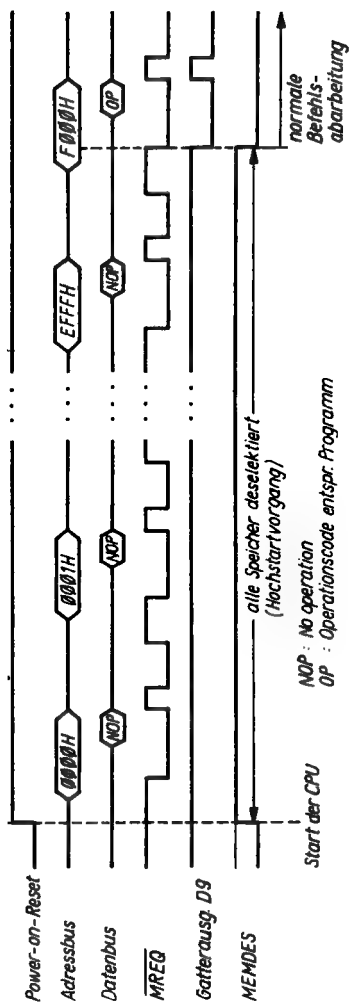


Bild 3.3 Impulsdiagramm Startlogik

senen Speicherschaltkreise zu deselektieren (siehe Abschnitt 3.3.1.).

3.4. Aufbau und Inbetriebnahme

Aufbau und Inbetriebnahme der CPU-Baugruppe sollen mit einfachen, auch dem Amateur zugänglichen Hilfsmitteln realisiert werden. Es müßte jedoch ein Oszilloskop – möglichst als Zweistrahlenausführung – für diesen Zweck verfügbar sein.

Die CPU-Platine kann man dann wie folgt systematisch bestücken und erproben:

1. Zunächst werden nur der Quarzoszillator einschließlich 1:4-Untersetzer und Puffergatter sowie die RESET-Logik auf der Leiterplatte bestückt. CPU und EPROM sind nicht auf die Fassungen gesteckt. Außer den genannten sind keine weiteren Schaltkreise eingelötet.

Mit dem Oszilloskop ist am Oszillatorausgang die Frequenz von 10 MHz und am Pin 6 der CPU-Fassung die Rechteckspannung mit der Frequenz von 2,5 MHz zu kontrollieren.

2. Danach bestückt man die Startlogik mit den Schaltkreisen D 6, D 7, D 8, D 9 und den Widerständen R 9 . . . R 16. Die Funktion der RESET-Schaltung und der Startlogik kann nun zusammen kontrolliert werden.

Nach Einschalten der Betriebsspannung + 5 V bewirkt das RESET-Signal am Eingang von Gatter D 6.2 das Setzen des RS-Flip-Flop, alle Ausgänge der Open-Collektor-Gatter D 7, D 8 müssen bei ordnungsgemäßer Funktion L-Signal führen. Beim Anlegen eines H-Signals an die Eingänge für AB 12 . . . AB 15 von D 9 und eines L-Signals an den Gattereingang D 6.1 muß das RS-Flip-Flop zurückkippen, und die Gatterausgänge von D 7, D 8 müssen auf H liegen.

3. Einlöten der Bustreiber 8212 und 8216 sowie des Schaltkreises D 4. Anschließend wird der Flip-Flop-Eingang D 6.3 vom Gatterausgang D 9 getrennt und auf H gelegt. Damit ist ein Zurückkippen des RS-Flip-Flop nach Erreichen der Adresse F000H beim Startvorgang verhindert. Zuletzt steckt man die CPU auf die Fassung. Nach Einschalten der Betriebsspannung und dem folgenden RESET werden alle Datenleitungen DB0 . . . DB7 auf L-Potential gezogen. Durch die Blockierung des RS-Flip-Flop ist dieser Zustand permanent vorhanden.

Infolge dieser Maßnahme liest die CPU ausschließlich NOP-Befehle ein, und der gesamte Adreßbereich wird zyklisch durchlaufen. Mit dem Oszilloskop kontrolliert man jetzt die Adreßleitungen AB0 ... AB15. Die an AB0 anliegende Pulsfolge weist die höchste Frequenz auf, die an AB15 die niedrigste. In ansteigender Reihenfolge der Adressen muß an der jeweiligen Adreßlinie die halbe Frequenz der vorhergehenden anliegen. Obwohl die Impulsfolge an den einzelnen Adreßausgängen wegen des Refresh-Zyklus unterbrochen ist, muß der beschriebene Sachverhalt deutlich auf dem Oszilloskop erkennbar sein. Auch an den Steuersignalausgängen \overline{RD} , \overline{MREQ} und $\overline{M1}$ muß eine Impulsfolge anstehen.

4. Nun wird noch die Chipselektierung für die Speicherschaltkreise der CPU-Baugruppe überprüft.

Dazu lötet man als nächstes die Schaltkreise D 17 und D 5 ein. Die Datenleitungen bleiben, wie unter 3. beschrieben, auf L-Potential. Der Blockselektieingang C des MH 7442 (D 17) wird vom Ausgang des Gatters D 9 getrennt und auf Masse gelegt. Nach dem Einschalten der Betriebsspannung muß ein Impuls an den \overline{CS} -Eingängen der EPROM-Speicher (Pin 20) bzw. des RAM-Blocks (Pin 13) anliegen. Mit dem Zweistrahloszilloskop kann überprüft werden, ob die 4 \overline{CS} -Signale – wie es notwendig ist – zeitlich versetzt auftreten. Man darf natürlich nur jeweils 1 Speicherschaltkreis (bzw. RAM-Block) gleichzeitig selektieren.

5. Zum Abschluß werden alle noch fehlenden Schaltkreise und Bauelemente eingelötet. Die Blockierung des RS-Flip-Flop der Startlogik wird aufgehoben, und alle Verbindungen gemäß Schaltung in Bild 3.1 werden wieder hergestellt.

Wer jetzt über die Möglichkeit verfügt, einen PROM programmieren zu lassen (was er später zur Inbetriebnahme des Monitor-Programms ohnehin muß), kann die Ein-/Ausgabefunktion noch testen.

Dazu programmiert man einen EPPROM mit:

```
M1:   IN A
      OUT A
      INC C
      JR M1
```

Der EPROM wird auf den Steckplatz 1 der CPU-Platine gesteckt und das kleine Programm mit Power-On-RESET und Startlogik bei F000H gestartet. Auf den Steuerleitungen \overline{IORQ} , \overline{RD} und

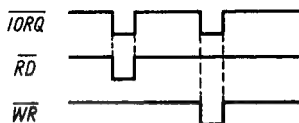


Bild 3.4 Impulsdiagramm Ein-/Ausgabefunktion

\overline{WR} muß der in Bild 3.4 dargestellte Impulsverlauf nachweisbar sein.

Die Ausgänge der I/O-Portdecodierung müssen I/O-Selektsignale aufweisen.

Diesen Zeitverlauf kontrolliert man mit dem Zweistrahloszilloskop.

Der Test ermöglicht gleichzeitig noch einmal den Nachweis für die einwandfreie Funktion der Startlogik.

Nach positivem Abschluß der Inbetriebnahme gemäß 1. bis 5. sind wesentliche Funktionen der CPU-Baugruppe überprüft. Man kann jetzt mit hoher Wahrscheinlichkeit davon ausgehen, daß ihre ordnungsgemäße Arbeitsweise abgesichert ist. Eine Gesamtüberprüfung stellt dann noch einmal die später folgende Inbetriebnahme des Monitorprogramms im Zusammenhang mit den peripheren Baugruppen dar.

Es soll hier aber auch nicht unerwähnt bleiben, daß manchmal bestimmte versteckte Fehler erst nach längerer Betriebszeit entdeckt werden.

4. Speichererweiterung

Wie bereits erwähnt, stellt die in Abschnitt 3. beschriebene CPU-Baugruppe einen in sich funktionsfähigen Einplatinenrechner dar. Zu einem leistungsfähigen Mikrocomputersystem gehört selbstverständlich mehr, vor allem eine ausreichend aufgerüstete Speicherbaugruppe.

Im Zusammenhang mit der Startlogik wurde darauf hingewiesen, daß im vorliegenden Systemkonzept ein möglichst großer, zusammenhängender RAM-Bereich bevorzugt wird. Voraussetzung für die Abarbeitung häufig benötigter Programme im RAM-Bereich ist aber ihr vorheriges Laden von einem externen Speichermedium (Kassette, Lochband, Floppy-Disk) bzw. von einem Festwertspeicher (PROM bzw. EPROM). Es ist leicht einzusehen, daß es nicht möglich ist, beispielsweise einen 8 . . . 12 kByte langen BASIC-Interpreter jedesmal vor einer gewünschten Benutzung über die Tastatur einzugeben.

Weiterhin ist es sinnvoll, ein häufig benötigtes Programm, wie z. B. den erwähnten Monitor, direkt auf PROM-Speicher laufen zu lassen. Voraussetzung ist natürlich, daß er dafür geeignet ist.

Im folgenden werden die Schaltungen für die zur Speichererweiterung notwendigen ROM- und RAM-Platinen angegeben und diskutiert. Als RAM-Speicher werden je eine Schaltungsvariante auf der Basis des statischen RAM *U 202* sowie des für einen effektiven Speicherausbau günstigen dynamischen RAM *U 256 (4116)* vorgestellt.

An dieser Stelle ist es erforderlich, auf das Problem der Pufferung der Busleitungen auf Speicher- und Peripheriebaugruppen kurz einzugehen.

Der Datenverkehr im Computer ist grundsätzlich bidirektional. Das bedeutet also, daß alle Datenquellen (außer der CPU auch Speicher und Peripherie) in der Lage sein müssen, den Datenbus mit allen angeschlossenen Systemkomponenten hinsichtlich Lastfaktor und kapazitiver Belastung zu treiben.

Aus der Sicht der CPU ist dies durch die eingesetzten Treiberschaltkreise abgesichert. Für Speicher- und Peripheriebaugruppen in größeren Mikrocomputersystemen können sich aber in diesem

Zusammenhang Probleme ergeben. Besonders beachten muß man die Kapazitäten der Ein- und Ausgänge von Speicherschaltkreisen, die sich durch die übliche Parallelschaltung der Adreß- und Datenleitungen entsprechend vergrößern.

Aus den genannten Gründen ist es bei allgemein verwendbaren OEM-Computern wie z. B. *K 1520* (OEM: *Original Equipment Manufactures*) sowie in großen Systemen üblich und notwendig, die Leitungen auf der jeweiligen Speicher- bzw. Peripheriebaugruppe zu puffern. Aus Gründen des Schaltkreisbedarfs ist es im vorliegenden System vertretbar, auf diese Pufferung zu verzichten. Dabei muß man darauf achten, daß die Belastung des Busses kleiner als eine TTL-Last bleibt und sich die oben erwähnten Lastkapazitäten in annehmbaren Grenzen halten.

Wer über ausreichend Treiberschaltkreise verfügt, sollte die Busleitungen auf den Speicherplatinen (z. B. PROM mit 8212, RAM mit 8216) bzw. Peripheriebaugruppen jedoch puffern.

4.1. ROM-Speicher

Bild 4.1 zeigt die Schaltung für eine 8-kByte-Prom-Leiterplatte auf der Basis des Schaltkreises *U 555* (2708).

Mit AB0 ... AB9 werden die 1k 8-Bit-Speicherzellen auf dem jeweiligen Chip adressiert. Das Chipselektsignal für die 8 auf der Leiterplatte platzierten PROM wird durch 1-aus-8-Decodierung der Adressen AB10, AB11, AB12 sowie des in Abschnitt 3.3.1. erläuterten Blockselektsignals der zentralen Speicherselektierung erzeugt. Für diesen Zweck nutzt man den 1-aus-10-Decoder *MH 7442*. Dem Problem direkt angepaßt ist natürlich der eventuell schwieriger beschaffbare 1-aus-8-Decoder 8205.

Die Pufferung der Datenausgänge bleibt, wie oben ausgeführt, dem einzelnen überlassen. Falls sie gepuffert werden sollen, eignet sich dazu der Schaltkreis 8212. Dieser Schaltkreis bietet sich deshalb an, weil vom PROM-Speicher selbstverständlich nur Daten gelesen werden können (der Fall des PROM-Programmierens ist in diesem Zusammenhang nicht von Interesse) und damit nur Datenfluß in der einen Richtung PROM → CPU möglich ist.

Mit H-Pegel am STR-Eingang wird die 8-Bit-Information der jeweiligen Speicherzelle des gerade selektierten PROM in den D-Latches des 8212 gespeichert. Wenn $\overline{DS1} = L$ (entsprechend

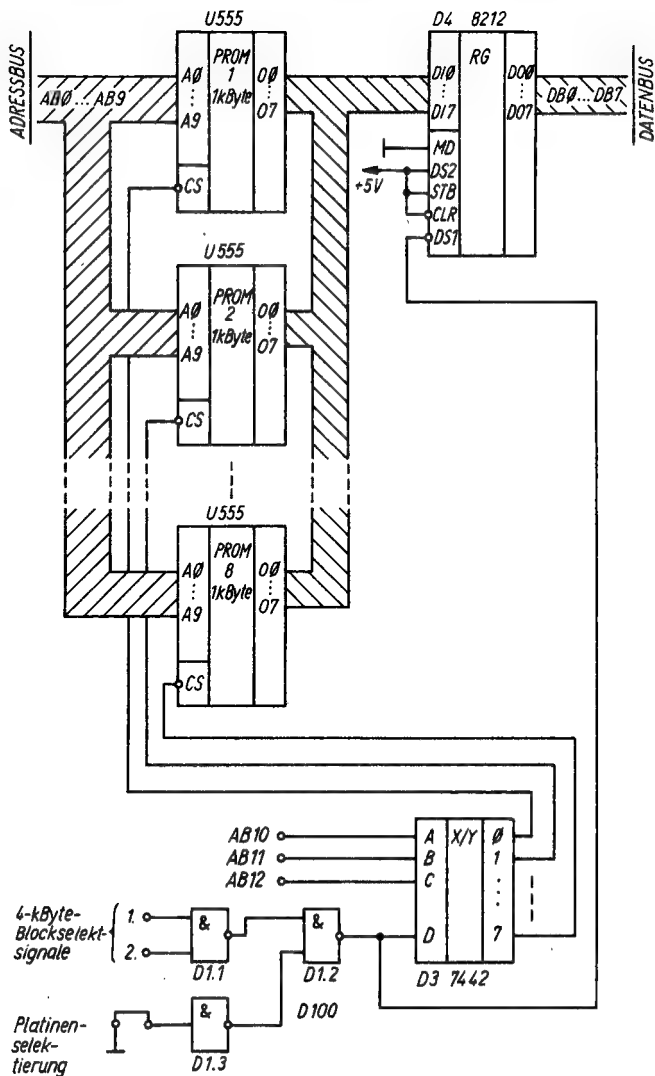


Bild 4.1 8-kByte-PROM-Platine mit U555

dem Blockselektsignal) und $DS2 = H$, liegt die Information an den Tri-State-Ausgängen des 8212 und damit gepuffert am Datenbus. Die zentrale Speicherselektierung liefert 4-kByte-Blockselektsignale, von denen für eine 8-kByte-Platine natürlich 2 Stück benötigt werden. Im allgemeinen schließt man 2 adressenmäßig benachbarte Blockselektsignale an die Eingänge von Gatter D1.1 an und erhält damit einen durchgängigen 8-kByte-Prom-Speicher. Wenn es sinnvoll erscheint, kann man diesen Speicher auch in zwei 4-kByte-Blöcke unterteilen.

Die Selektierung bzw. Deselektierung der gesamten Speicherplatine wird über Gatter D1.3 vorgenommen. Liegt der Gattereingang auf Masse (siehe Bild 4.1), ist die Platine selektiert.

Nachdem die Speichereinheit auf der Basis von $1k \times 8$ -organisierten PROM vorgestellt wurde, seien noch einige Hinweise gegeben, die es gestatten, die Platine so zu gestalten, daß sie auch für künftig verfügbare Speicherschaltkreise mit entsprechend größeren Speicherdichten verwendbar sind. Dies wurde möglich, weil es für alle nach dem Byte-wide-Prinzip (8 Bit Speicherbreite) organisierten Speichertypen eine standardisierte Pin-Belegung gibt. Diese Pin-Belegung gilt für die PROM-Typen *U 555 (2708)*, *2716*, *2732*, *2764* und ist dem Anhang zu entnehmen. Man muß beachten, daß der 8-kByte-Speichertyp (z. B. *2764*) 28polig ausgeführt ist.

International existieren bereits RAM-Speicher [z. B. *6116* (Hitachi), *4118* (MOSTEK)], die zu den genannten PROM-Typen pin-kompatibel sind. Damit wäre es sogar möglich, eine entsprechend gestaltete Leiterplatte wahlweise als PROM- und/oder als RAM-Platine zu verwenden. Beim Einsatz von RAM-Schaltkreisen ist es natürlich notwendig, die CPU-Signale \overline{RD} und \overline{WR} mit auf die Leiterplatte zu führen und für die Lese/Schreib-Umschaltung heranzuziehen.

Bild 4.2 zeigt die Schaltung einschließlich Verdrahtungsprinzip für eine universell verwendbare Speicherplatine.

Die Adreßeingänge A10, A11, A12 sowie die \overline{CS} -Eingänge der Speicherschaltkreise werden ebenso wie die Eingänge des Chipselektdecoders *MH 7442* als unbelegte Wickel- oder Lötanschlüsse auf der Platine ausgeführt. Sie können damit durch Drahtbrücken so beschaltet werden, wie es der jeweilige Speichertyp erfordert.

Die Tabelle 4.1. enthält die notwendige Adreßbelegung für vier 8-Bit-organisierte Speichertypen. Die Verdrahtung für den 1-kByte-Speicher (z. B. *U 555*) entspricht der in Bild 4.1, wobei man 2 Ein-

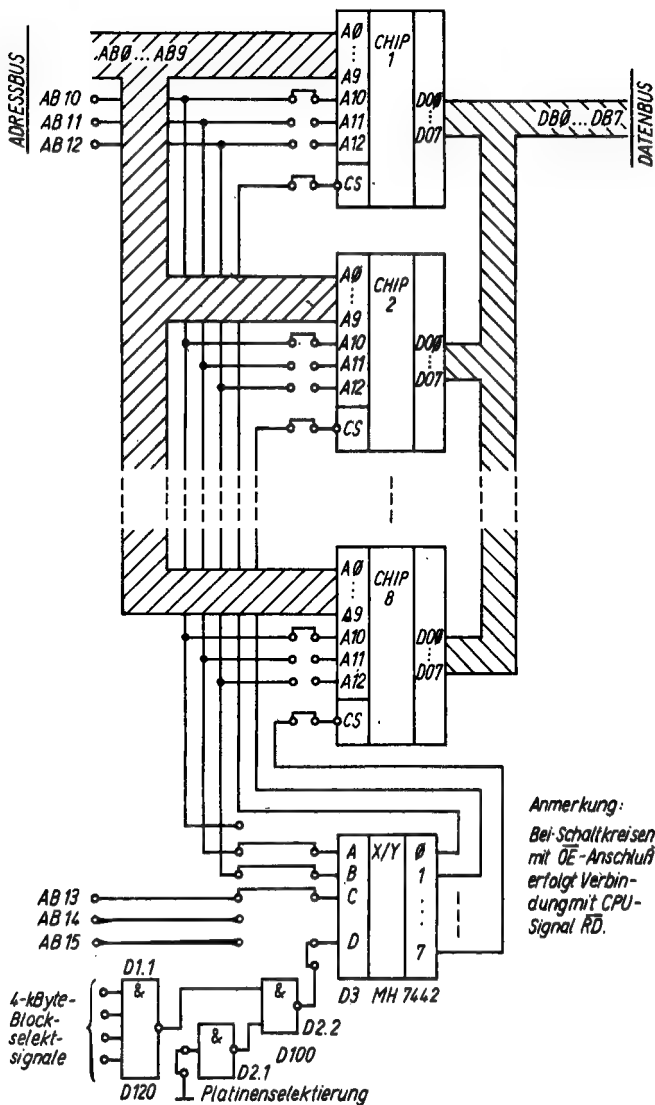


Bild 4.2 Universell verwendbare Platine für byte-wide-organisierte Speichertypen

Tabelle 4.1. Adreßbelegung für universell verwendbare Speicherplatine

Adreßbus	SPEICHERTYP			
	U 555 (1 kByte)	2716 (2 kByte)	2732 (4 kByte)	2764 (8 kByte)
AB0...AB9	A0...A9	A0...A9	A0...A9	A0...A9
AB10	A } Ein-	A10	A10	A10
AB11	B } gänge	A } Ein-	A11	A11
AB12	C } MH7442	B } gänge	A } Ein-	A12
AB13	/ ~	C } MH7442	B } gänge	A } Ein-
AB14	/	/	C } MH7442	B } gänge
AB15	/	/	/	C } MH7442

gänge des Gatters D 1.1 in Bild 4.2 auf H-Pegel legen muß. Beim 2-kByte-Speichertyp (z. B. 2716) sind 4 (im allgemeinen benachbarte) 4-kByte-Blockselektsignale der zentralen Speicherselektierung an die Eingänge von Gatter D 1.1 anzuschließen. Für den 4-kByte-Speichertyp (z. B. 2732) wird der Chipselektdecoder nicht benötigt. Die Ausgänge des *MH 74154* der zentralen Speicherselektierung werden direkt mit den CS-Eingängen der 8 Speicherschaltkreise verbunden. Für den 8-kByte-Speicher (z. B. 2764) ist die zentrale Speicherselektierung also gegenstandslos, da 8 Chips in diesem Fall den gesamten Adreßbereich der CPU *U 880* ausfüllen. Dementsprechend ist der Eingang D des *MH 7442* auf Masse zu legen, wobei der Gatterausgang von D 2.2 natürlich vorher abgetrennt werden muß.

4.2. RAM-Speicher

4.2.1. Statischer RAM-Speicher

Das Schaltungsprinzip für eine Speicherplatine auf der Basis des statischen RAM *U 202* wurde bereits im Zusammenhang mit dem auf der CPU-Baugruppe untergebrachten Speicher in Abschnitt 3.2. (Bild 3.1) beschrieben. Der Aufbau einer 4-kByte-RAM-Platine, bei der der Treiberschaltkreis 8212 verwendet wird, ist damit sinngemäß möglich. Die Chipselektierung, d. h. in diesem Fall das Selektieren der vier 1-kByte-RAM-Blöcke, sowie die Lese/

Schreib-Umschaltung sind identisch. Die 3 PROM ersetzt man durch 1-kByte-RAM-Blöcke und faßt die Datenausgänge DO0 . . . DO7 entsprechend ihrer Wertigkeit zusammen. Nach dem Puffern durch den 8212 werden sie auf den Datenbus geschaltet.

Bild 4.3 zeigt die Datenpufferung mit 2 bidirektionalen Treiberschaltkreisen. Wie ebenfalls bereits in Abschnitt 3.3. erwähnt wurde, ist (bei Vorhandensein der benötigten Treiber 8216) dieser Variante der Vorzug zu geben.

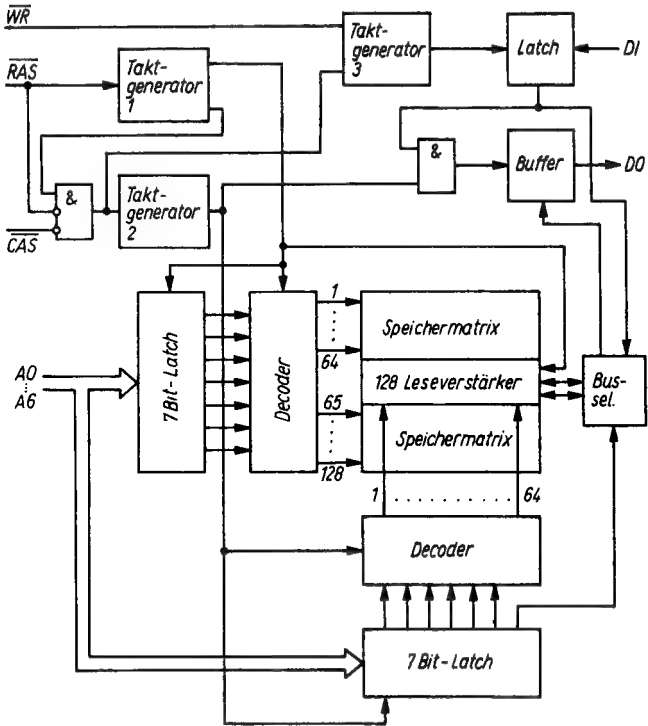
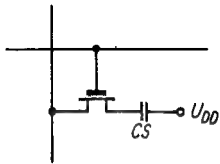
4.2.2. Dynamischer RAM-Speicher

Dynamische RAM haben gegenüber statischen RAM einige Vorteile: So werden in einem Schaltkreis weitaus mehr Speicherzellen vereinigt, die Leistungsaufnahme ist bedeutend geringer, und der Preis je Bit ist kleiner.

Bisher wurden nur statische Speicherschaltkreise verwendet, denn sie lassen sich leicht handhaben, und sie benötigen keine externe Steuerung. Da es kaum Timing-Probleme gibt, läßt sich ein Computer mit statischen RAM wesentlich einfacher in Betrieb nehmen. Die Autoren empfehlen den Bau dieser dynamischen RAM-Karte deshalb als Erweiterung. Die Inbetriebnahme ist für den Amateur sonst eventuell zu schwierig.

Im Gegensatz zu statischen RAM, in denen bekanntlich Flip-Flop die Informationen speichern, enthalten die modernen dynamischen RAM Eintransistorzellen (Bild 4.4a). Gespeichert wird in der Kapazität C_s . Diese Speicherzellen gestatten hohe Integrationsdichten, d. h., es lassen sich sehr viele Speicherplätze auf einem Chip unterbringen.

Für die dynamische RAM-Karte wurde der 16384×1 -Bit-RAM U256 (4116) verwendet. Die Funktionsweise eines dynamischen 16-kBit-RAM soll kurz an Hand von Bild 4.4b erläutert werden. Die einzelnen Speicherzellen sind in einer Matrix von 128 Zeilen (row) und 2×64 Spalten (column) angeordnet. Die 14-Bit-Adresse wird dem RAM zu je 7 Bit multiplex als Zeilen- und Spaltenadresse zugeführt. Die beiden Adressen werden in zwei 7-Bit-Latches übernommen: Die Zeilenadresse durch einen \overline{RAS} -Impuls (*row address strobe*), die Spaltenadresse durch einen \overline{CAS} -Impuls (*column address strobe*). Wie bereits erwähnt, wird die Information in Kapazitäten gespeichert, die sich durch Leckströme langsam entla-



b)

Bild 4.4 a – dynamischer RAM U 256 (4116), Eintransistorspeicherzelle, b – Übersichtsplan

den. Um einen Informationsverlust zu vermeiden, muß jede Zeile mindestens alle 2 ms aufgefrischt werden (*refresh*). Das ist übrigens eine beachtlich lange Zeit, wenn man bedenkt, daß die Kapazität einer Speicherzelle nur etwa 0,04 pF beträgt!

Beim Lesen und Auffrischen verstärken 128 Leseverstärker die Ladung der Kapazitäten. Die Leseverstärker sorgen für ein Nachladen der Kapazitäten und beschreiben während des Schreibvorgangs ($\overline{WE} = L$) die Zellen neu.

Der dynamische 16-kBit-RAM ermöglicht verschiedene Betriebsarten. Gesteuert wird über die Eingänge \overline{RAS} , \overline{CAS} und \overline{WE} . Bild 4.5 zeigt Impulsdiagramme für die verwendeten Betriebsarten. Die angegebenen Zeiten sind typische Werte (abhängig von der Zugriffszeit und vom Hersteller).

Die in Bild 4.6a gezeigte Schaltung realisiert die Ansteuerung für maximal 64-kByte-RAM. Der Decoder D1 (8205) wählt einen der 16-k-Blöcke aus. Der Schaltkreis D2 wird als Latch genutzt. Er verhindert Konflikte, die sich durch vorzeitigen Adreßwechsel ergeben können [2]. Ein Speicherzugriff ($\overline{MREQ} = L$, $\overline{RFSH} = H$) gibt den Decoder frei. Einer der Decoderausgänge wird aktiv (L-Pegel). Der entstehende Impuls steht hinter den Gattern D 4.1 ... D 4.4 als $\overline{RAS1}$, $\overline{RAS2}$, $\overline{RAS3}$ oder $\overline{RAS4}$ bereit. An den Adreßeingängen des Speicherschaltkreises liegen die 7 niederwertigen Adreßleitungen. Nachdem der RAM die Zeilenadresse übernommen hat, kann der Multiplexer (D8 und D9) umschalten und die 7 höherwertigen Adreßleitungen an den RAM legen. Liegt die Adresse sicher am RAM an, wird sie durch \overline{CAS} übernommen. Die Verzögerungskette ist für das Timing verantwortlich. Um die Early-Write-Betriebsart zu realisieren, wird das \overline{CAS} -Signal nochmals verzögert (Gatter D6.2, D6.3, D7.3, D7.4). Das \overline{CAS} -Signal steuert außerdem die Tri-State-Treiber.

Während eines Refreshzyklus ($\overline{MREQ} = L$, $\overline{RFSH} = L$) ist der Decoder D1 inaktiv. Am Ausgang des Gatters D3.3 liegt L-Pegel, so daß über D4.1 – D4.4 alle RAS-Leitungen L-Pegel erhalten. Es wird ein „RAS-ONLY“-Zyklus ausgelöst (eine Zeile der Matrix wird aufgefrischt). Bild 4.7 soll die Vorgänge verdeutlichen.

Die kleine Logik in Bild 4.6b mit den Gattern D13.1, D13.2 und D5.2 hat die Aufgabe, die oberen 6 kByte der 64-kByte-RAM-Karte auszublenden. Der Bereich von E8000H ... FFFFH wird bereits von den Speichern auf der CPU-Karte und dem Bildwiederholpeicher belegt.

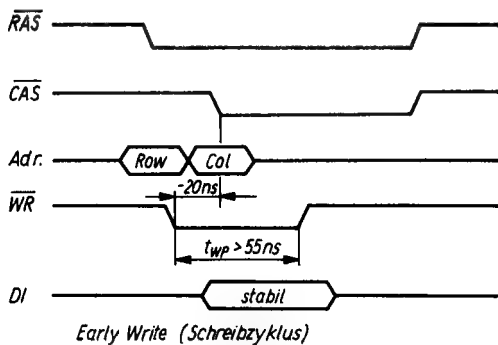
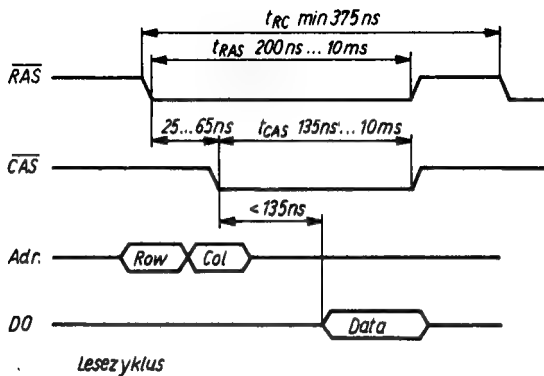
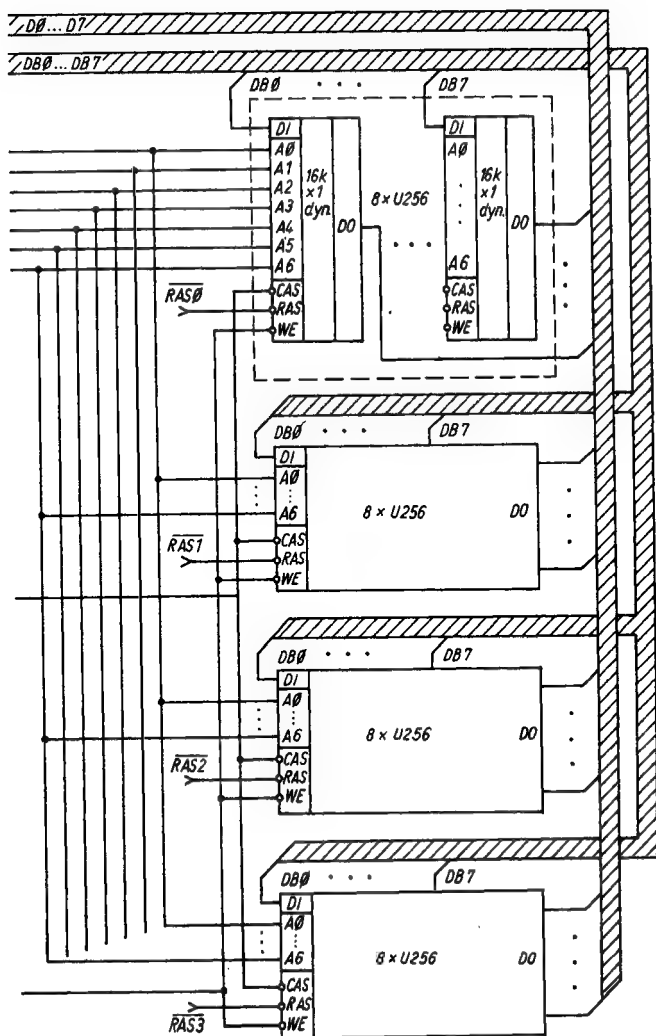
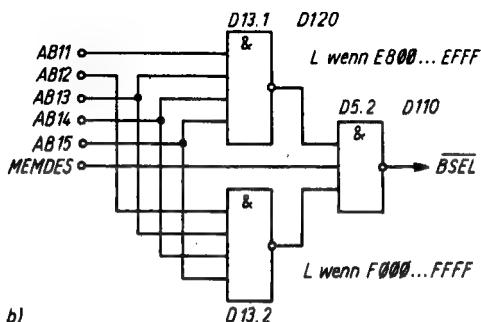


Bild 4.5 Impulssdiagramm der dynamischen RAM U 256 (4116)





b)

Bild 4.6 a – 64-kByte-RAM-Speicherbaugruppe mit U256 (4116), b – Logik zum Ausblenden des Speicherbereichs E800H...FFFFH

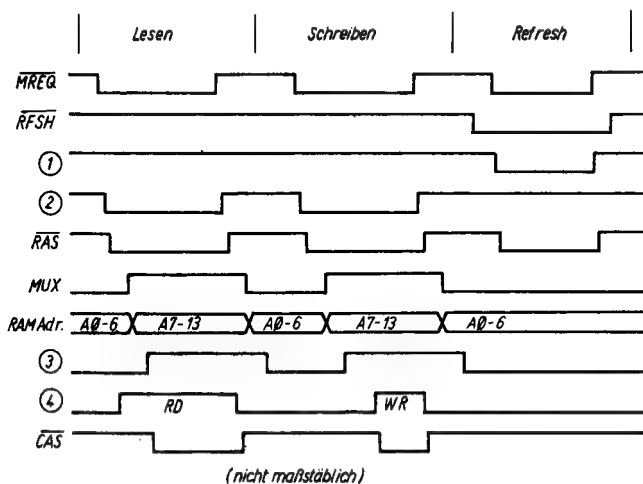


Bild 4.7 Impulsdiagramm der 64-kByte-RAM-Speicherbaugruppe mit U256 (4116)

4.3. Aufbau und Inbetriebnahme

Die Inbetriebnahme der PROM-Platine und des statischen RAM-Speichers ist bei sauberem, übersichtlichem Aufbau problemlos.

Für den Einsatz des EPROM U 555 (2708) sowie des dynamischen RAM U 256 (4116) ist jedoch folgendes zu beachten:

Um ein Zerstören der genannten Speichertypen auszuschließen, muß die Betriebsspannung $U_{BB} = -5\text{ V}$ beim Einschalten zuerst und beim Ausschalten zuletzt am Schaltkreis anliegen (Einzelheiten dazu im Abschnitt 10.1.).

Die weiteren Hinweise zu Aufbau und Inbetriebnahme beschränken sich auf den dynamischen RAM-Speicher. Wie bereits oben erwähnt, ist der mittlere Leistungsverbrauch eines dynamischen 16-kBit-RAM nur gering: im aktiven Zustand etwa 420 mW, im Standby-Zustand maximal 20 mW. Die mittleren Betriebsströme betragen: $I_{DD} = 35\text{ mA}$, $I_{CC} = 10\text{ }\mu\text{A}$, $I_{BB} = 200\text{ }\mu\text{A}$. Wenn der RAM jedoch angesteuert wird, entstehen Spitzenströme bis zu 100 mA je RAM. Deshalb ist jeder Schaltkreis besonders an den Anschlüssen U_{DD} und U_{BB} mit Stützkondensatoren (je 100 nF) abzublocken. Die Leitungen für die Betriebsspannung und die Masse müssen ausreichend dimensioniert werden.

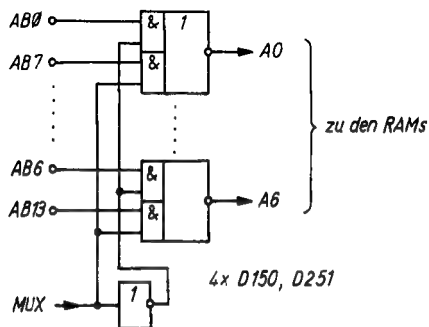
Die DIL-Schalter (oder Drahtbrücken) S1 . . . S4 ermöglichen das Ein- und Ausschalten einzelner 16-k-Blöcke. Man sollte zunächst nur einen 16-k-Block mit Speicherschaltkreisen bestücken. Wenn sich die Speicherzellen nicht beschreiben lassen oder gar ihre Information verändern, so werden mit einem Oszilloskop die Impulsbilder überprüft. Vor allen Dingen bei unterschiedlich schnellen Speichern ist ein wenig Fingerspitzengefühl beim Einstellen der Verzögerungszeiten notwendig. Die RC-Werte sind erprobt für folgende RAM-Typen:

MK 4116-3, D 416C-2 und TMM 416P-3.

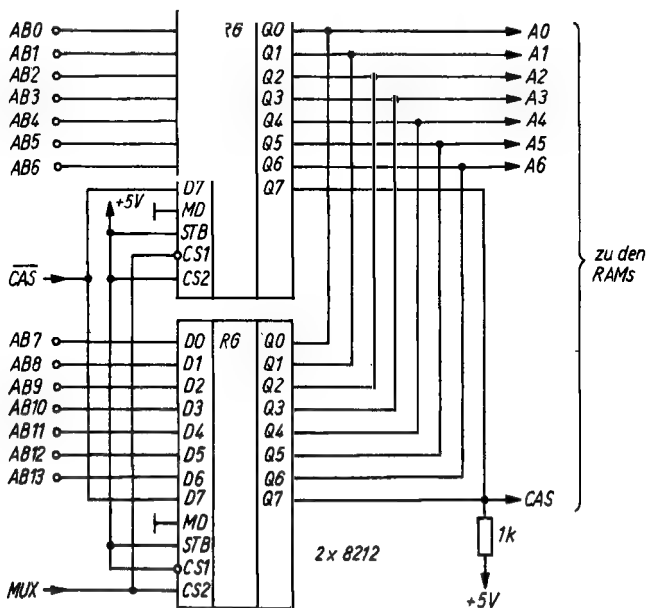
War das Schreiben und Lesen erfolgreich, dann sollte man einen RAM-Test mit dem in Tabelle 4.2. dargestellten Programm (nach

Tabelle 4.2. HEXA-Tabelle RAM-Testprogramm

Adresse HEXA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FC00	06	00	21	00	ST	7D	AC	A8	77	23	7C	FE	SP	C2	05	FC
FC10	21	00	ST	7D	AC	A8	BE	C4	25	FC	23	7C	FE	SP	C2	13
FC20	FC	04	C3	02	FC	C5	0E	05	CD	21	F0	52	41	4D	20	46
FC30	65	68	6C	65	72	20	41	64	72	2E	00	0E	04	CD	21	50
FC40	0E	05	CD	21	F0	1E	00	C1	C9	ED	FF	52	41	4D	54	45
FCS0	53	54	00	C3	00	FC										



a)



b)

Bild 4.8 Adreßmultiplexer; a – mit D 150, b – mit 8212

MOSTEK) anschließen. Das Programm arbeitet mit dem in Abschnitt 9. beschriebenen Monitor zusammen. Einzutippen ist das Programm ab Adresse FC00H. ST bedeutet H-Teil der Anfangsadresse des zu testenden Speicherbereichs, SP bedeutet H-Teil der Endadresse. Die 4 Byte muß man selbst eintragen.

Gestartet wird das Testprogramm ab Adresse FC00H oder durch Eingabe des Befehls RAMTEST. Der Test eines 16-kByte-Speichers dauert etwa 4 Minuten. Der Speicher läßt sich beliebig lange testen (Abbruch durch $\overline{\text{RESET}}$ oder $\overline{\text{NMI}}$). Alle fehlerhaften Speicheradressen werden ausgegeben. Auch wenn dieser Test erfolgreich war, ist noch nicht hundertprozentig sicher, daß der Speicher funktioniert. Es kann sein, daß die Zugriffszeit der RAM-Karte während des $\overline{\text{M1}}$ -Zyklus nicht klein genug ist. Dann müssen die Zeitkonstanten der $\overline{\text{RC}}$ -Glieder der Verzögerungskette geringfügig verkleinert werden.

Ein Hinweis noch zu den verwendeten Bauelementen: Als Multipler wurden Schaltkreise vom Typ 74157 eingesetzt.

Bild 4.8 zeigt 2 Alternativen. Um den Datenbus nicht zu belasten, benutzt man Treiberschaltkreise vom Typ 8216.

5. Fernsehinterface

Die Kommunikation zwischen Mensch und Computer setzt geeignete Ein-/Ausgabemedien voraus. Von besonderer Bedeutung in diesem Zusammenhang sind Geräte bzw. Systemkomponenten, die den visuell unterstützten Dialog ermöglichen. Aus der Vielzahl der Displayvarianten hat sich als universelles Anzeigemedium für Ziffern- und Zeichendarstellung bis hin zur Graphik der Bildschirm durchgesetzt. Das gilt in besonderem Maße für den Bereich der Mikrocomputer. Die Industrie bietet hierfür spezielle Bildschirmgeräte (vom Schwarzweiß- bzw. einfarbigen Monitor bis hin zum Farbmonitor) an:

Für viele Belange, vor allem natürlich aus der Sicht des Amateurs, erweist sich für den genannten Zweck der Einsatz eines handelsüblichen Fernsehgeräts als technisch und ökonomisch sinnvoll. Im folgenden wird ein relativ komfortables, problemlos nachbaubares Fernsehinterface beschrieben.

5.1. Zeichendarstellung auf dem Bildschirm

Das vom Fernsehen bekannte Darstellungsprinzip beruht auf der Helligkeitssteuerung des Elektronenstrahls in der Bildröhre. Der abhängig von der Videoinformation getastete Strahl überstreicht dabei in einem Zeilenraster den Bildschirm von oben nach unten. Das geschieht nach CCIR-Norm mit einer Zeilenfrequenz von 15625 Hz (d. h. einer Zeilendauer von $64 \mu\text{s}$) und 625 Zeilen je Bild. Daraus ergibt sich eine Bildwechselfrequenz von 25 Hz. Durch den Trick des Zeilensprungverfahrens erzeugt man eine Rasterwechselfrequenz von 50 Hz. Durch das Verkoppeln der Bildfrequenz wird ein flimmerfreies Bild abgesichert.

Zum Darstellen der Zeichen auf dem Bildschirm sollte man auf das Zeilensprungverfahren verzichten und statt dessen 2. identische Halbbilder mit etwa der Hälfte der 625 Fernsehzeilen erzeugen. Damit ist die für eine ausreichende Flimmerfreiheit des Bildes notwendige Bildfrequenz von etwa 50 Hz garantiert.

Die Anzahl der Fernsehzeilen (gewählt wurden 320 je Bild) ermög-

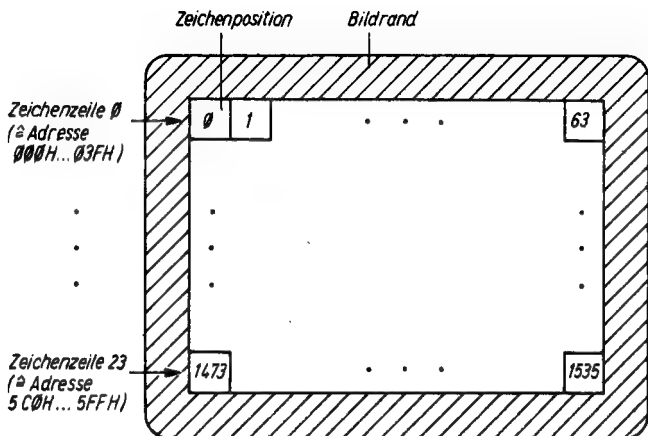


Bild 5.1 Bildschirmformat

licht die Darstellung einer relativ großen Anzahl von Zeichen auf dem Bildschirm (einschließlich des oberen und des unteren Bildrands).

Bild 5.1 zeigt das verwendete Bildschirmformat. Ein Bild besteht danach aus 24 Zeichenzeilen, jede Zeichenzeile aus 64 Zeichenpositionen. Damit ist die Darstellung von maximal 1536 Zeichen (d. h. 1,5 k) je Bild möglich.

Mit dieser Lösung wurde ein sinnvoller Kompromiß zwischen den technischen Möglichkeiten eines handelsüblichen TV-Geräts und der Forderung nach einer Abbildungsmöglichkeit für eine ausreichend große Zeichenzahl in gut lesbarer Form gefunden. Dabei nimmt man bewußt in Kauf, daß 0,5 kByte des 2-kByte-Bildwiederholerspeichers nicht ausgenutzt sind.

Jeder Zeichenposition auf dem Bildschirm ist eine Adresse im Bildwiederholerspeicher in steigender Reihenfolge fest zugeordnet. Der Bildschirmposition 0 entspricht also die Anfangsadresse CRTBG, der Position 1535 die Adresse CRTBG + 1535 des Bildwiederholerspeichers.

Ein Zeichen wird auf dem Bildschirm nach dem häufig benutzten Prinzip der 5 × 7-Punktmatrix aufgebaut. Bild 5.2 zeigt die Zeichenzusammensetzung für das verwendete Bildschirminterface auf der Grundlage dieses Prinzips. Man benötigt zum Darstellen der

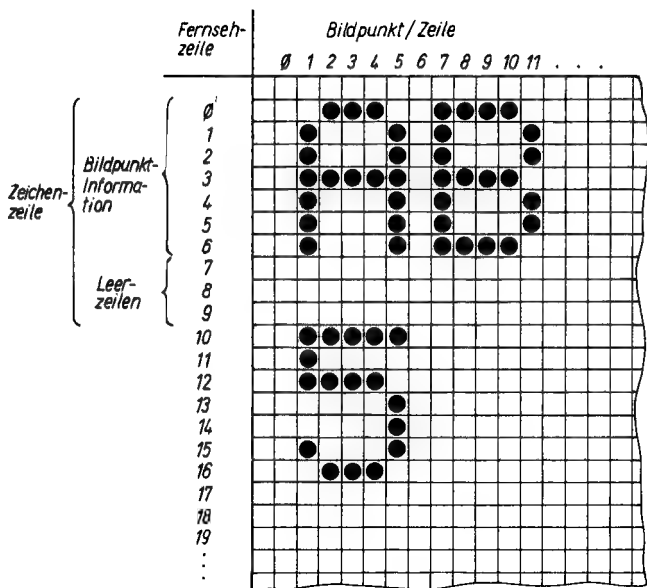


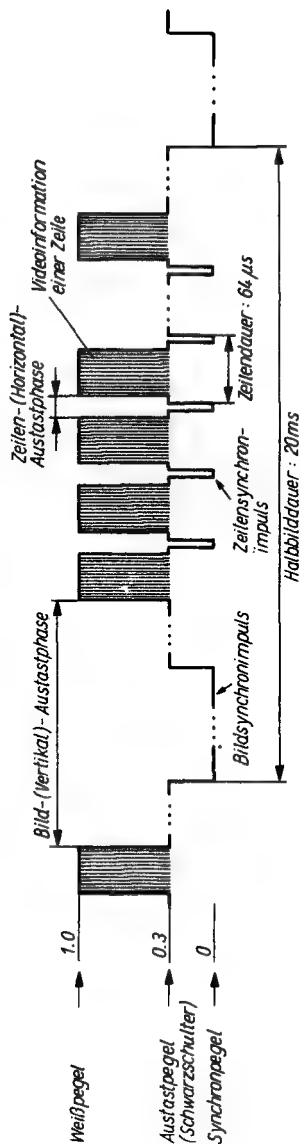
Bild 5.2 Zeichenaufbau nach dem Prinzip der 5 × 7-Punktmatrix

Information einer Zeichenzeile also 7 Fernsehzeilen. Zur Gewährleistung eines übersichtlichen, gut lesbaren Bildes wird jede Zeichenzeile durch 3 weitere Fernsehzeilen von der benachbarten getrennt. Insgesamt besteht damit eine Zeichenzeile aus 10 Fernsehzeilen.

Für jedes Einzelzeichen stehen in der Horizontalen, d. h. je Fernsehzeile, 6 Bildpunkte zur Verfügung. Jeweils der 1. oder 6. Bildpunkt (abhängig von der gewählten Bitposition auf dem Zeichengenerator) wird dunkelgetastet und bildet den horizontalen Zeichenabstand.

Unter Berücksichtigung von Bildschirmformat und Zeichenaufbau setzt sich das Bildfeld damit aus 240 Fernsehzeilen zusammen. Eine Zeile besteht aus 384 Bildpunkten.

Bild 5.3 zeigt das BAS-Signal (Bild-, Austast- und Synchronsignal) nach CCIR, das mit dem Bildschirminterface erzeugt werden muß, um ein handelsübliches TV-Gerät ansteuern zu können. Danach



Darstellung vereinfacht, nicht maßstäblich

Bild 5.3 BAS-Signal nach CCIR

ist neben der Videoinformation, die die 384 Bildpunkte je Zeile abhängig von dem jeweils darzustellenden Zeichen hell- und dunkeltastet, auch die Bereitstellung der Zeilen- und Bildsynchrosignale sowie der entsprechenden Austastsignale notwendig.

Durch das Zeilenaustasten entstehen ein linker und ein rechter Bildrand. In die Zeilenaustastphase fällt auch der Strahlrücklauf, der damit nicht sichtbar ist. Bei der Bildaustastung entstehen ein oberer und ein unterer Bildrand.

Durch die Festlegung von Bildrändern wird abgesichert, daß alle 24 Zeichenzeilen zu je 64 Zeichen eindeutig erkennbar sind. Es werden also keine Zeichen von den Bildschirmrändern „verschluckt“. Die an den Bildschirmrändern häufig auftretenden Unschärfen und Verzeichnungen haben keine Auswirkungen.

5.2. Schaltungsbeschreibung

Bild 5.4 zeigt einen Übersichtsschaltplan, an dem das grundsätzliche Funktionsprinzip des Fernsehinterface des Mikrocomputers erklärt werden soll.

Der Zugriff der CPU zum Bildschirm ist über den Bildwiederhol-speicher sichergestellt, da dieser wie ein normaler RAM-Speicher (siehe Abschnitt 4.2.) verwaltet wird. In diesem Fall sind die Adreßleitungen des BildwiederholSpeichers über den Adreßmulti-plexer auf den Systembus geschaltet.

Die CPU kann jetzt mit einem normalen Speicherschreibzyklus den Zeichencode (ISO-7-Bit bzw. ASCII) auf der gerade gewählten Adresse im BildwiederholSpeicher ablegen. Damit ist, wie bereits in Abschnitt 5.1. erwähnt, auch die Zeichenposition auf dem Bildschirm festgelegt.

In dieser Art wird der gesamte auf dem Bildschirm darzustellende Bildinhalt im BildwiederholSpeicher abgespeichert.

Nachdem der Zugriff der CPU auf den BildwiederholSpeicher beendet ist, schaltet der Adreßmultiplexer um, und die umfangreiche Elektronik der Interfaceschaltung sorgt dafür, daß der vollständige Inhalt des BildwiederholSpeichers in eine zum Tasten des Elektronenstrahls geeignete Videoinformation umgewandelt und zyklisch mit der Bildfrequenz wiederholt wird. Ein Taktgenerator liefert den Bildpunkt-takt und steuert damit eine Zählerkette an. Diese Zählerkette wurde so gestaltet, daß die entsprechenden Zähler-

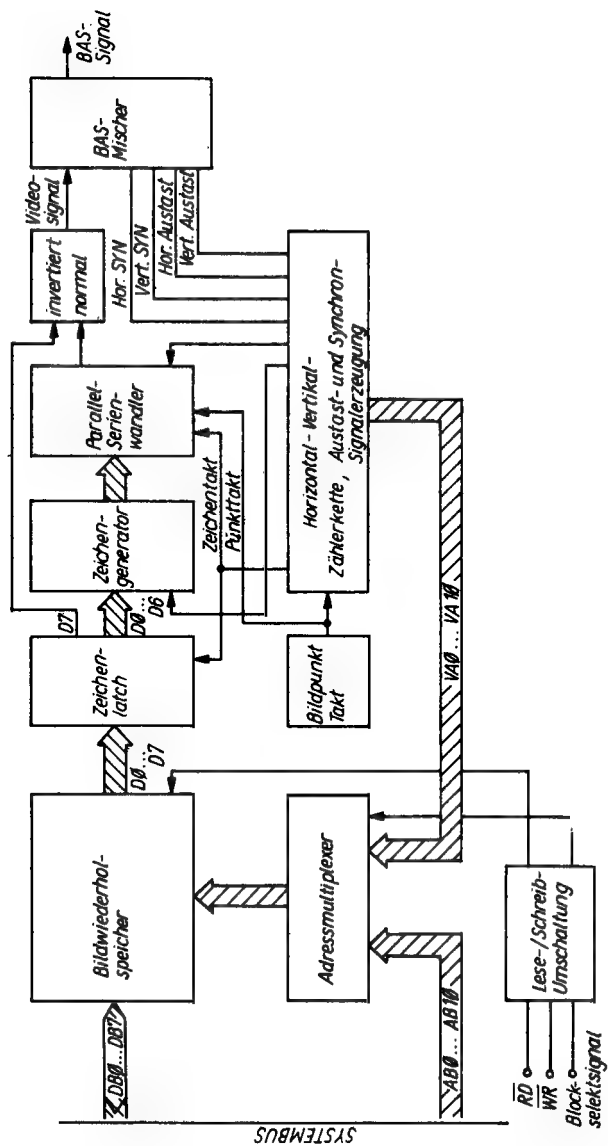


Bild 5.4 Übersichtsschaltplan des Fernsehinterface

ausgänge über den Adreßmultiplexer direkt zur zyklischen Adressierung des BildwiederholSpeichers herangezogen werden können. Diese Adreßinformation wird synchron mit der dazugehörigen Position des Elektronenstrahls auf dem Bildschirm bereitgestellt. Aus den Zählerzuständen gewinnt man außerdem durch geeignete logische Verknüpfung die Zeilen- und Bildsynchronimpulse sowie die dazugehörigen Austastsignale.

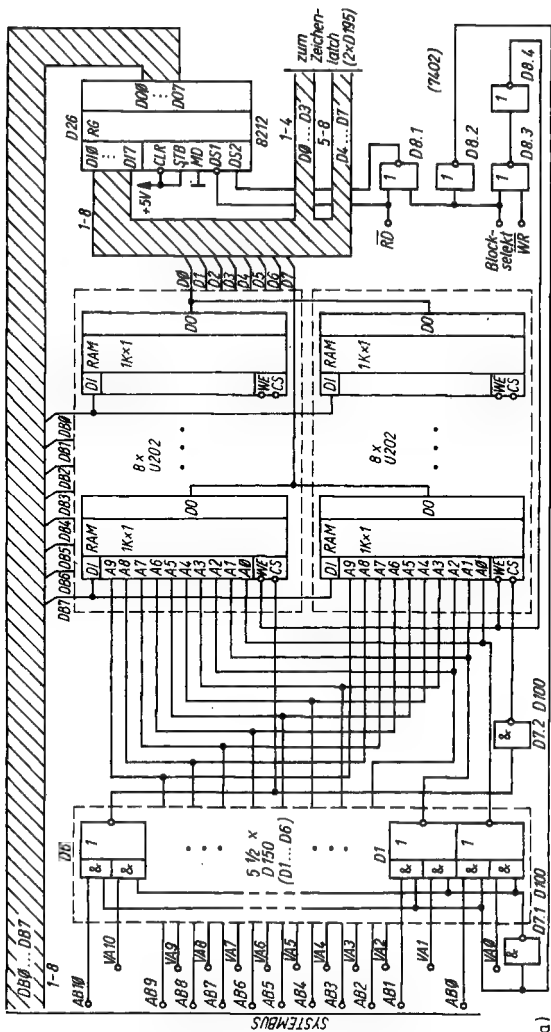
Die gerade adressierte Speicherzelle im BildwiederholSpeicher stellt den aktuellen Zeichencode über den Zeichenlatch dem Zeichengenerator zur Verfügung. Im Zeichengenerator, einem Festwertspeicher (PROM), sind die der jeweiligen Zeichenkonfiguration (siehe Bild 5.2) entsprechenden Bildpunktinformationen abgespeichert und liegen an den Ausgängen parallel in 8-Bit-Breite vor. Am Ausgang des anschließenden Parallel-Serienwandlers läßt sich jetzt die serielle Videoinformation abnehmen. Sie wird danach mit dem zur Zeichenauswahl (maximal 128 verschiedene Zeichen) nicht benötigten Bit 8 des BildwiederholSpeichers exklusiv – oder verknüpft. Damit ist die Zusatzfunktion „Zeicheninvertierung“ möglich.

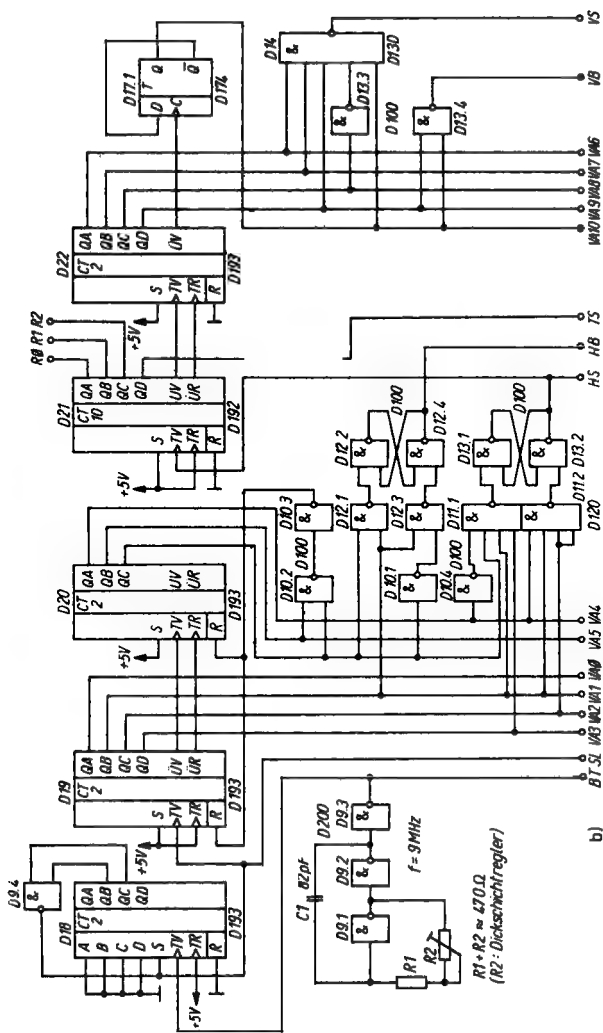
Synchrone Austast- und Videosignale werden am BAS-Mischer zusammengeführt und zu dem in Bild 5.3 dargestellten BAS-Signal nach CCIR-Norm verarbeitet.

5.2.1. BildwiederholSpeicher, Adreßmultiplexer

Bild 5.5 zeigt die gewählte Schaltung für das Bildschirminterface. Den BildwiederholSpeicher muß man so konzipieren, daß in ihm der gesamte Bildinhalt gespeichert werden kann. Im vorliegenden Beispiel sind das die Codes für 1536 Zeichen, d. h. 1,5 kByte.

Der BildwiederholSpeicher ist wieder mit statischen RAM U 202 aufgebaut. Auf Grund seiner 1024×1 -Bit-Organisation sind also in der bereits bekannten Weise 2 RAM-Blöcke zu je 1 kByte Speicherkapazität zu realisieren. Damit bleiben (wie bereits erläutert wurde) 0,5 kByte ungenutzt. Prinzipiell ließe sich dieser Rest natürlich als normaler RAM-Speicher nutzen. Zu empfehlen ist dies aber nicht, da sich (ohne besondere schaltungstechnische Maßnahmen) jeder CPU-Zugriff zum BildwiederholSpeicher durch einen kurzzeitigen undefinierten Bildschirmzustand bemerkbar macht. Der würde natürlich auch beim Ansprechen des 0,5-kByte-Rests auftreten, obwohl sich am Bildinhalt selbst nichts änderte.





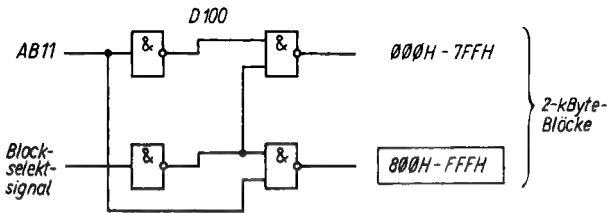


Bild 5.6 Selektierung des BildwiederholSpeichers

In Abhängigkeit davon, ob die CPU Zugriff zum Bildschirminterface hat oder nicht, müssen die Adreßleitungen A0 ... A10 des BildwiederholSpeichers entweder auf den Datenbus des Systems oder auf die zur zyklischen Adressierung notwendigen Zählerausgänge gelegt werden. Dieses Umschalten bewirkt ein Adreßmultiplexer, der mit Schaltkreisen des Typs D 150 aufgebaut wurde (D1 ... D6). Eleganter wäre der Einsatz von 3 Schaltkreisen vom Typ 74157. Dieser Schaltkreis beinhaltet jeweils vier 2-auf-1-Multiplexer [3].

Über die NOR-Gatter von D6 wird der BildwiederholSpeicher bei E800H (siehe Bild 3.3) selektiert und der Adreßmultiplexer umgeschaltet.

An Stelle des Treiberschaltkreises 8212 eignen sich selbstverständlich auch 2 Exemplare des Typs 8216.

Das 4-kByte-Blockselektsignal (E000H) wird wieder der zentralen Speicherselektierung entnommen, wobei man entsprechend der Speicherkapazität des BildwiederholSpeichers und der gewählten Adresse E800H noch in zwei 2-kByte-Blöcke unterteilen muß. Dies ist durch 1-aus-2-Decodierung des Blockselektsignals und des Adreßbit AB11 möglich (Bild 5.6).

5.2.2. Synchron- und Austastsignalerzeugung, Adressenbereitstellung

Den Bildpunkttakt liefert ein freischwinger Impulsgenerator, bestehend aus den schnellen Gattern des Typs D 200 (D 9.1 ... D 9.3). Als Taktfrequenz wurde $f_{BP} = 9 \text{ MHz}$ ausgewählt. Damit ergeben sich günstige Zeitverhältnisse.

Bildpunktdauer: $t_{BP} = 111 \text{ ns}$; mit 64 darstellbaren Zeichen je Zei-

chenzeile und 6 Bildpunkten je Zeichen und Fernsehzeile erhält man 384 Bildpunkte je Fernsehzeile. Daraus folgt eine genutzte Zeilendauer von $384 \cdot 111 \text{ ns} = 42,6 \mu\text{s}$. Das bedeutet also, daß 42,6 μs der verfügbaren Zeilendauer von 64 μs für das Schreiben einer Zeile des Bildfelds genutzt werden. Der verbleibende Rest von 21,4 μs wird dunkel getastet und ergibt den rechten und den linken Bildrand.

Mit dem Bildpunkttakt wird ein schneller 6 : 1-Zähler auf der Basis des 4-Bit-Binärzählers *D 193* (D18) angesteuert. Er zählt die 6 Bildpunkte, die je Fernsehzeile zum Darstellen eines Zeichens notwendig sind, und gibt danach den Impuls an den folgenden Zeichenpositionszähler ab. Dieser besteht aus 2 Schaltkreisen *D 193* (D19, D20) und zählt die Zeichenpositionen je Zeichenzeile bis 64. Die Gatter D10.2 und D10.3 übernehmen das Rücksetzen dieses Zählers nach 96 Eingangsimpulsen (d. h. 576 Bildpunktimpulsen).

Aus den Zählerzuständen des Zeichenpositionszählers werden in geeigneter Form der Zeilensynchronimpuls (Gatter D10.4, D11, D13.1, D13.2) und der Zeilenaustastimpuls (Gatter D12, D10.1) decodiert. Für den Zeilensynchronimpuls ergeben sich eine Impulsbreite von $t_{\text{IH}} = 8 \mu\text{s}$ und die geforderte Periode von $T_{\text{H}} = 64 \mu\text{s}$. Die Austastzeit beträgt, wie bereits gezeigt, $t_{\text{AH}} = 21,4 \mu\text{s}$. Bild 5.7 zeigt die zeitlichen Abläufe.

Man erkennt die um 2 Zeichenpositionen (12 Bildpunkte entsprechend 1,33 μs) verzögerte Helltastung. Damit werden keine undefinierten Zeichen auf dem Bildschirm sichtbar.

Der Ausgangsimpuls der bisher beschriebenen Horizontalzählerkette, d. h. der Zeilensynchronimpuls, gelangt auf den Eingang der sich anschließenden Vertikalzählerkette. Der am Beginn dieser Zählerkette befindliche Dezimalzähler *D 192* (D21) erhält also mit jeder Fernsehzeile, die der Elektronenstrahl auf den Bildschirm geschrieben hat, einen Ansteuerimpuls. Er zählt die zur Darstellung einer Zeichenzeile notwendigen 10 Fernsehzeilen (7 für die eigentliche Zeicheninformation, 3 für den vertikalen Zeichenzeilenabstand).

Nach Beendigung dieses Vorgangs wird ein Impuls an den nachfolgenden Zeichenzeilenzähler abgegeben. Dieser besteht aus dem Schaltkreis *D 193* (D22) sowie einem Binäruntersetzer *D 174* (D17.1) und zählt bis 32. Da nach 32 Eingangsimpulsen der Zählvorgang wieder bei 0 beginnt, ist kein Rücksetzen erforderlich.

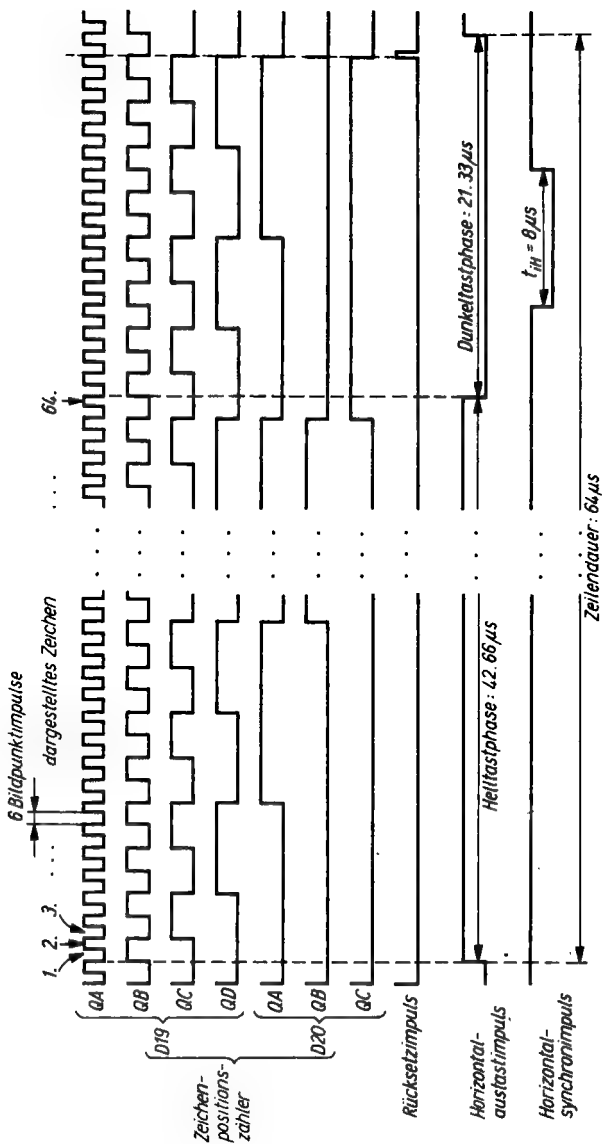


Bild 5.7 Impulsdiagramm der Horizontalablenkung

Das gesamte Bild besteht damit aus 32 Zeichenzeilen, entsprechend also 320 Fernsehzeilen. Davon entfallen 24 Zeichenzeilen auf das Bildfeld, die verbleibenden 8 (d. h. 80 Fernsehzeilen) bilden den oberen und den unteren Bildrand. Es ergibt sich daraus eine Bildfrequenz von

$$\frac{1}{320 \cdot 64 \mu s} = 48,8 \text{ Hz.}$$

Damit ist eine ausreichende Flimmerfreiheit des Bildes gewährleistet.

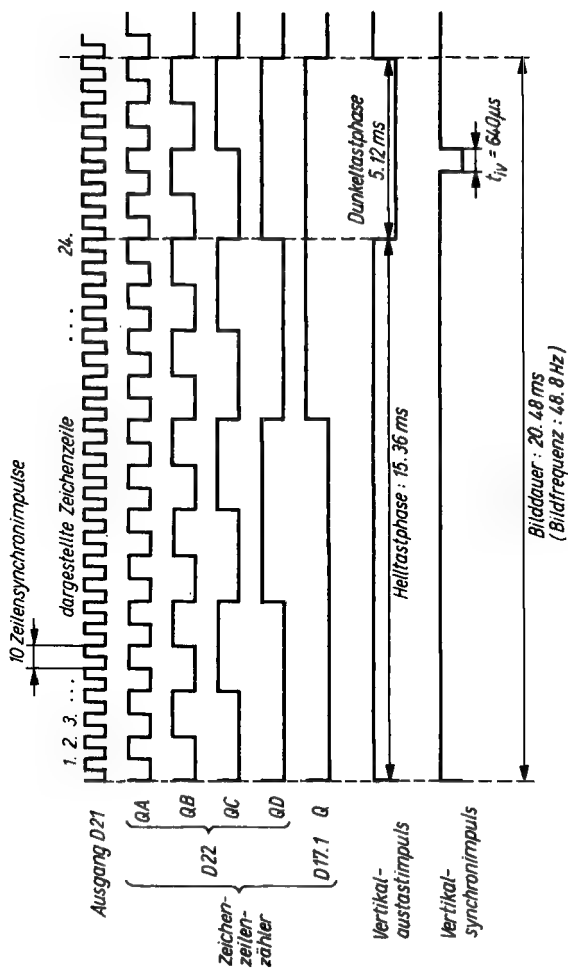
Aus den Zählerzuständen des Zeichenzeilenzählers werden mit der Gatterkombination D13.3, D14 der Vertikalsynchronimpuls ($t_{IV} = 640 \mu s$) sowie der Vertikalaustastimpuls ($t_{AV} = 80 \cdot 64 \mu s = 5,12 \text{ ms}$) mit dem Gatter D13.4 decodiert.

Bild 5.8 zeigt die Zeitverhältnisse für die Vertikalimpulserzeugung.

Über die zyklische Adressierung durch die entsprechenden Zählerausgänge werden dem Bildwiederholpeicher die *Zeichenposition* (0 . . . 63) und die *Zeichenzeile* (0 . . . 23) signalisiert, in der sich der Elektronenstrahl gerade befindet. Genau aus diesem Sachverhalt ergibt sich die eindeutige Zuordnung zwischen Zeichenpositionen auf dem Bildschirm und Adresse im Bildwiederholpeicher. Die Ausgänge QA, QB, QC des Dezimalzählers D 192 sind mit den Adreßeingängen A0, A1, A2 des im weiteren noch zu beschreibenden Zeichengenerators verbunden. Damit wird dem Zeichengenerator mitgeteilt, in welcher Fernsehzeile des gerade darzustellenden Zeichens sich der Elektronenstrahl befindet. Der Ausgang QD des D 192 sorgt über den Taktsperr Eingang des Parallel-Serienwandlers 74165 dafür, daß in den letzten 2 Fernsehzeilen (d. h. während der Zählerzustände 8, 9) jeder Zeichenzeile keine Videoinformation ausgegeben wird. Diese beiden sowie eine weitere, auf dem Zeichengenerator erzeugte, Leerzeile trennen die Zeichenzeilen in der Vertikalen.

5.2.3. Videosignalerzeugung

Der aktuelle Zeichencode, d. h. der Inhalt der gerade im Bildwiederholpeicher adressierten Speicherzelle, wird mit jedem Ausgangsimpuls des Bildpunktzählers in den Zeichenlatch übernommen und steht gleichzeitig an dessen Ausgang zur Verfügung.



Der Zeichenlatch wurde mit zwei 4-Bit-Schieberegistern des Typs *D 195* aufgebaut. Er stellt dem Zeichengenerator unter Berücksichtigung der Speicherzugriffszeiten den Zeichencode als stabile Adresse zur Verfügung.

Der Zeichencode bildet also über die Adreßeingänge A3 ... A9 eine von 128 möglichen Adressen für den Zeichengenerator-PROM. Auf genau dieser Adresse sind die Bildpunktinformationen für die 7 Zeilen des darzustellenden Zeichens in 7 aufeinanderfolgenden Speicherzellen als Datenbyte abgespeichert. Die 8. Speicherzelle in dieser Reihenfolge ist mit 00H belegt und bildet damit die 1. der 3 Leerzeilen für den vertikalen Zeichenzeilenabstand.

Es werden nur 6 Bit der 8-Bit-Speicherbreite des PROM genutzt (5 Bit Bildpunktinformation, 1 Bit horizontaler Zeichenabstand). Insgesamt benötigt man damit zum Darstellen eines Zeichens 8 aufeinanderfolgende Speicherzellen des Zeichengenerator-PROM. Die 8 Speicherzellen werden, wie in Abschnitt 5.2.2. bereits erwähnt, durch die Zählerausgänge QA, QB, QC des Dezimalzählers D 192 an den Adreßeingängen A0, A1, A2 des Zeichengenerators adressiert.

Tabelle 5.1. enthält die Bildpunktinformation für den Buchstaben

Tabelle 5.1. Zuordnung: Zeichencode – Speicherzelle – Bildpunktinformation auf dem Zeichengenerator-PROM

Zeichencode (A3 ... A9)	PROM-Speicherzelle (A0 ... A9)	Zeichen								Datenbytes
		D7							D0	
41H	208H			●	●	●				38H
	209H		●					●		46H
	20AH		●					●		44H
	20BH		●	●	●	●		●		7CH
	20CH		●					●		46H
	20DH		●					●		46H
	20EH		●					●		46H
	20FH									00H
										Leerzeile

↑
horizontaler Zeichenabstand
nicht belegt

Beispiel: Adressbitbelegung Zeichengenerator-PROM

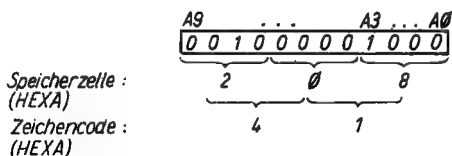


Tabelle 5.2. HEXA-Tabelle Zeichengenerator

0000H/03FFH

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000	00	08	18	38	78	38	18	08	00	28	28	28	28	28	28	28	0010	00	7C	00	44	28	10	28	44	00	00	00	7E	00	7E	00	00
00020	00	7C	00	58	64	44	44	44	00	00	00	34	48	48	34	34	0030	00	30	48	48	58	44	58	40	00	18	20	10	38	44	44	38
00040	00	00	00	00	10	28	44	7C	00	00	40	70	48	48	08	04	0050	00	38	44	44	7C	44	44	38	00	00	40	20	10	28	44	44
00060	00	00	00	24	24	38	40	00	00	04	38	68	28	28	28	28	0070	00	00	00	3C	48	48	48	30	00	7C	20	10	08	10	20	7C
00080	00	00	00	04	38	50	10	10	00	00	10	38	54	38	10	10	0090	00	38	44	44	44	44	28	60	00	10	00	38	44	7C	44	44
000A0	00	10	00	38	48	48	34	00	28	00	38	44	7C	44	44	44	00B0	00	28	00	38	48	48	34	00	28	38	44	44	44	44	44	38
000C0	00	28	00	38	44	44	44	38	00	28	00	44	44	44	44	38	00D0	00	00	28	00	44	44	44	38	00	00	10	08	7C	08	10	00
000E0	00	00	3C	20	20	60	20	00	30	48	10	20	78	00	00	00	00F0	00	18	24	20	70	20	20	7C	00	44	28	10	10	38	10	10
00010	00	00	00	00	00	00	00	00	00	20	20	20	20	20	20	20	0110	00	28	28	00	00	00	00	00	00	28	28	7C	28	7C	28	28
000120	00	10	3C	50	38	14	78	10	00	60	64	08	10	20	4C	0C	0130	00	20	50	20	54	48	34	00	30	30	10	10	20	00	00	00
000140	00	08	10	10	10	10	08	00	20	10	10	10	10	10	10	20	0150	00	00	10	54	38	54	10	00	00	10	10	7C	10	10	10	00
000160	00	00	00	00	30	30	10	20	00	00	00	00	7C	00	00	00	0170	00	00	00	00	00	30	30	00	00	04	08	10	20	40	00	00
000180	00	38	44	4C	54	64	44	38	00	10	30	10	10	10	10	38	0190	00	38	44	04	18	20	40	7C	00	38	44	04	18	04	44	38
0001A0	00	08	18	28	48	7C	08	00	7C	40	78	04	04	44	38	38	01B0	00	18	20	40	78	44	44	38	00	7C	04	08	10	20	20	20
0001C0	00	38	44	44	38	44	44	38	00	38	44	44	3C	04	08	30	01D0	00	00	30	30	00	30	30	00	00	30	30	30	30	10	20	10
0001E0	00	04	08	10	20	10	08	04	00	00	00	7C	00	7C	00	00	01F0	00	40	20	10	08	10	20	40	00	38	44	04	08	10	00	10
002000	00	38	44	5C	54	5C	40	38	00	38	44	44	7C	44	44	44	0210	00	78	44	44	78	44	44	78	00	38	44	40	40	44	44	38
002020	00	78	44	44	44	44	44	78	00	7C	40	40	78	40	40	7C	0230	00	7C	40	40	78	40	40	40	00	38	44	40	40	4C	44	3C
002040	00	44	44	44	7C	44	44	44	00	38	10	10	10	10	10	38	0250	00	04	04	04	04	04	44	38	00	44	48	50	60	50	48	44
002060	00	40	40	40	40	40	40	7C	00	44	6C	54	54	44	44	44	0270	00	44	44	64	54	4C	44	44	00	38	44	44	44	44	44	44
002080	00	78	44	44	78	40	40	40	00	38	44	44	44	54	48	34	0290	00	78	44	44	78	50	48	44	00	38	44	40	38	04	44	38
0020A0	00	7C	10	10	10	10	10	10	00	44	44	44	44	44	44	38	02B0	00	44	44	44	28	28	10	10	00	44	44	44	44	54	54	6C

Tabelle 5.2. HEXA-Tabelle Zeichengenerator

0000H/03FFH

ADDR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
02C0	00	44	44	28	10	28	44	44	00	44	44	28	10	10	10	10	02D0
02E0	00	00	40	20	10	08	04	02	00	70	10	10	10	10	10	70	02F0
0300	00	30	30	20	10	00	00	00	00	00	00	38	48	48	34	34	0310
0320	00	04	04	34	4C	44	44	3C	00	00	00	38	44	78	40	38	0330
0340	00	40	40	58	64	44	44	44	00	10	00	30	10	10	10	38	0350
0360	00	30	10	10	10	10	10	38	00	00	00	28	54	54	44	44	0370
0380	00	00	00	78	44	64	58	40	00	00	00	38	48	58	28	0C	0390
03A0	00	00	20	70	20	28	10	00	00	00	00	44	44	44	4C	34	03B0
03C0	00	00	00	44	28	10	28	44	00	00	00	44	28	10	10	20	03D0
03E0	00	10	10	10	00	10	10	10	00	20	10	10	08	10	10	20	03F0

A sowie die entsprechenden Datenbyte, die auf dem Zeichengenerator abzuspeichern sind. Auch die Zuordnung zwischen Speicherzellenadressierung (A0 ... A9) und Zeichencode (A3 ... A9) ist aus der Tabelle ersichtlich.

Als Zeichengenerator wird der 1-kByte-PROM *U 555* verwendet. Über dessen Adreßeingänge A3 ... A9 lassen sich also 128 Zeichen zu je 8 aufeinanderfolgenden Speicherzellen (das entspricht genau 1 kByte) auswählen.

Für die notwendigen Zeichencodvereinbarungen gilt der ISO-7-Bit- bzw. ASCII-Code (siehe Tabellen im Anhang). Die Auswahl der 128 Zeichen und das Programmieren des Zeichengenerators-PROM sind unter Berücksichtigung der gegebenen Hinweise (Zeichenaufbau, Adreßbelegung, Codevereinbarung) nach eigenem Geschmack möglich. Beispielsweise können selbst zu gestaltende Pseudographikelemente mit vorgesehen werden.

In Tabelle 5.2. ist ein bewährter Zeichengenerator in Form einer Hexatabelle angegeben. Er enthält neben den Ziffern, den großen und kleinen Buchstaben alle Umlaute, die gebräuchlichsten griechischen Buchstaben sowie alle aus der Sicht der Thematik relevanten Sonderzeichen.

Entsprechend der Adressierung durch den jeweiligen Zeichencode liegt also die aktuelle Bildpunktinformation einer Zeile des gerade darzustellenden Zeichens an den Ausgängen 00 ... 07 des Zeichengenerators *U 555* parallel vor. Der sich anschließende Parallel-Serienwandler wandelt diese Information in ein serielles Videosignal um, wie es zum Tasten des Elektronenstrahls benötigt wird. Zur Parallel-Serienwandlung wird das 8-Bit-Schieberegister *74165* verwendet [3]. Für die Übernahme der Parallelinformation vom Zeichengenerator (d. h. zum Laden des Schieberegisters) ist ein Nadelimpuls erforderlich, der in dem Moment an den Shift-Load-Eingang des *74165* geliefert wird, wenn der Elektronenstrahl die 6 Bildpunkte des vorhergehenden Zeichens in der jeweiligen Fernsehzeile auf den Bildschirm geschrieben hat.

Der benötigte Ladeimpuls steht am Ausgang des Bildpunktzählers zur Verfügung. Er wird durch Gatter D15.4 invertiert und zum Laden des Zeichenlatch *D 195* verwendet.

Die Videoinformation muß selbstverständlich synchron mit dem Bildpunkttakt seriell ausgegeben werden. Zu dem Zweck ist der Takteingang des Schaltkreises *74165* direkt mit dem Ausgang des Taktgenerators verbunden.

Die sinnvolle Verwendung des Taktsperreingangs am Schieberegister 74165 wurde bereits beschrieben.

Im Abschnitt 5.2. wurde darauf hingewiesen, daß das Bit 8 des BildwiederholSpeichers zur Auswahl der 128 auf dem Zeichengenerator programmierten Zeichen nicht benötigt wird. Man realisiert deshalb damit die Zusatzfunktion „Zeicheninvertierung“, indem das Bit 8 nach Zwischenspeicherung im D-Latch D 174 (D17.2) mit dem Videosignal exklusiv – oder verknüpft (Gatter D15.1, D16.1, D16.2) wird. Für den Fall, daß Bit 8 H-Pegel führt, ist eine invertierte Zeichendarstellung (dunkel auf hellem Grund) sichergestellt.

5.2.4. BAS-Signalerzeugung

Ein TV-Gerät läßt sich nur mit BAS-Signal nach CCIR gemäß Bild 5.3 ansteuern. Dieses erzeugt man, indem das Videosignal mit den vorher beschriebenen Synchron- und Austastsignalen zusammengeführt wird. Hierzu werden die Horizontal- und Vertikalsynchronsignale am Gatter D15.3 und die entsprechenden Austastsignale am Gatter D15.2 und verknüpft. An den Ausgängen der Open-Kollektor-Gatter D 103 (D16) liegen jetzt alle Signale in richtiger Polarität vor. Der BAS-Mischer, Transistor VT 1, mischt sie und liefert am Ausgang polaritäts- und potentialgerecht das BAS-Signal. Das BAS-Signal wird dem Video-Eingang des TV-Geräts zugeführt.

Für bestimmte Fernsehgeräte (z. B. *Combivision*) hat es sich als günstig erwiesen, Video- und Synchronsignale getrennt den entsprechenden Stufen im TV-Gerät zuzuführen.

Verwendet man einen geeigneten HF-Modulator, kann das BAS-Signal auch direkt über den Antenneneingang eingespeist werden. Hierbei sind die Störstrahlungsbestimmungen der Deutschen Post zu beachten!

5.3. Erweiterungsmöglichkeiten

Das Fernsehinterface kann durch die mögliche Darstellung graphischer Informationen (z. B. in Verbindung mit den in Abschnitt 7. vorgestellten AD-Wandlern) auf dem Bildschirm sinnvoll erweitert werden. Dazu bieten sich mehrere Möglichkeiten.

5.3.1. Zeichengenerator mit Graphiksymbolen

Auf dem Zeichengenerator werden neben dem ASCII- bzw. ISO-7-Bit-Zeichensatz Graphikelemente (z. B. Blockgraphik) untergebracht. Diese Blockelemente gestaltet man so, daß sich aus ihnen die gewünschten graphischen Darstellungen auf dem Bildschirm zusammensetzen lassen. Als Zeichengenerator eignet sich ein PROM mit einer Speicherkapazität ≥ 2 kByte (z. B. 2716, 2732). Dabei wird der Ausgang QD des Zählers D 192 (D21) vom Taktsperreingang des Parallel-Serienwandlers 74165 (D25) abgetrennt und zum Adressieren des PROM herangezogen. Jetzt können alle 10 Fernsehzeilen einer Zeichenposition zur Informationsdarstellung genutzt werden (z. B. auch bei Buchstaben zum Erzeugen von Unterlängen).

5.3.2. RAM-Speicher als Zeichengenerator

Als Zeichengenerator wird ein RAM-Speicher verwendet. Die gewünschten Graphiksymbole (z. B. Kurvenabschnitt u. ä.) lassen sich jetzt nach Bedarf programmieren. Zu diesem Zweck werden die Datengänge der RAM (ggf. über Treiberschaltkreise) auf den Datenbus gelegt. Die Datenausgänge sind unmittelbar mit den Eingängen des Schieberegisters 74165 verbunden.

Die Adreßeingänge werden über 2-auf-1-Multiplexer bei CPU-Zugriff auf den Adreßbus und andernfalls auf die Latchausgänge D 195 (D23, D24) bzw. die Zählerausgänge QA, QB, QC des D 192 (D21) gelegt.

Den Zeichengenerator kann man auch in kombinierter RAM- bzw. PROM-Ausführung realisieren. Beispielsweise ließ sich der ASCII-Zeichensatz auf PROM abspeichern, und Graphikelemente in der oben geschilderten Art könnte man im RAM-Speicher programmieren.

Für den Taktsperreingang des 74165 gilt das in Abschnitt 5.3.1. Gesagte.

5.3.3. Vollgraphik

In diesem Fall kann jeder auf dem Bildschirm darstellbare Bildpunkt softwaremäßig angesprochen werden. Der Zeichengenera-

tor entfällt. Als Bildwiederholpeicher nutzt man einen RAM-Block mit entsprechend großer Speicherkapazität (z. B. 256×256 Bit). Hierzu wird entweder dem Fernsehinterface ein Bildwiederholpeicher zugeordnet oder ein Teil des Arbeitsspeichers des Computers genutzt. Im letzteren Fall liegen die Eingänge des Parallel-Serienwandlers dann direkt am Datenbus. Die Videoadressen (d. h. die Zählerausgänge zur zyklischen Adressierung) werden über Bustreiber an den Adreßbus gelegt. Die für den Speicherzugriff nötigen Steuersignale muß eine entsprechende Logik erzeugen. Der DMA-Vorgang, d. h. das Auslesen des Bildwiederholpeichers, liegt in der Vertikalhellastphase. Das Vertikalaustastsignal ist gleichzeitig das BUSRQ-Signal (in der Darstellung gemäß Bild 5.8 in invertierter Form). Die CPU kann damit nur in der Vertikalaustastphase arbeiten.

Mit dem Signal $\text{BUSAK} = L$ werden die Adreßtreiber freigegeben und damit die aktuelle CRT-Adresse auf den Adreßbus gelegt. Verwendet man dynamische RAM, dann muß das Refresh aufrechterhalten werden.

5.4. Aufbau und Inbetriebnahme

Bei Inbetriebnahme des Bildschirminterface werden zunächst die RAM-Bestückung des Bildwiederholpeichers und die CPU-Baugruppe ausgeklammert.

Man beginnt mit dem Einstellen des Taktgenerators auf 9 MHz. Damit wäre bereits der einzig notwendige Abgleichvorgang an der Schaltung erledigt. Hierzu noch ein Hinweis: Der Bildpunktakt wird mit dem Oszilloskop zweckmäßig durch exaktes Einstellen der Periode des Zeilensynchronimpulses mit $T_H = 64 \mu s$ fein justiert.

Anschließend werden sämtliche Synchron- und Austastsignale und, wenn diese in Ordnung sind, das komplette BAS-Signal oszilloskopisch kontrolliert. Genügen diese Signale den angegebenen Bedingungen (vergleiche Bild 5.3, Bild 5.7, Bild 5.8), legtman die Multiplexerausgänge $A_0 \dots A_7$ auf die entsprechenden Eingänge des Zeichenlatch. Auf dem Bildschirm muß jetzt der komplette Zeichensatz abgebildet sein (normal und invertiert).

Nach positivem Abschluß dieser Kontrollvorgänge werden die Verbindungen Multiplexer – Zeichenlatch beseitigt und der Bild-

wiederholungspeicher mit den RAM-Schaltkreisen bestückt. Im Anschluß hieran verbindet man die Bildschirmplatine über den Systembus mit der CPU-Baugruppe und hat jetzt die Möglichkeit, ein geeignetes Testprogramm zu starten.

Beispielsweise kann der Zeichensatz auf den Bildschirm geschrieben werden. Das kleine Programm dazu ist im folgenden angegeben:

```
LD HL, E800H
M1: LD (HL), L
    LD L, (HL)
    INC HL
    LD A, H
    CMP F0H
    JRNZ M1
    HALT
```

Das Bildschirminterface funktioniert dann vorschriftsmäßig, wenn der vollständige Zeichensatz 6mal in steigender Reihenfolge der Zeichen gemäß ISO-7-Bit bzw. ASCII auf dem Bildschirm geschrieben wird.

6. Alphanumerische Tastatur

Für den Dialog mit dem beschriebenen Mikrocomputer ist eine geeignete Tastatur die wichtigste Voraussetzung der Eingabe. Die im folgenden vorgestellte, einfach nachzubauende alphanumerische Tastatur läßt sich mit bis zu 64 Tasten auslegen. Im Gegensatz zu anderen denkbaren und auch üblichen Lösungen wird der jeweilige Tastaturcode hardwaremäßig erzeugt.

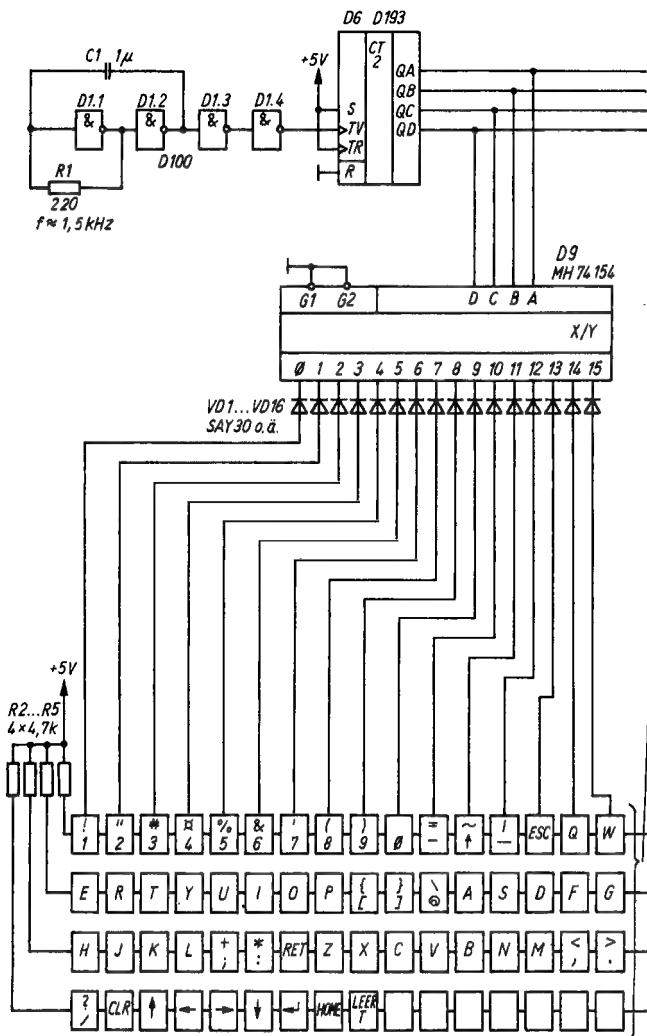
6.1. Schaltungsbeschreibung

Die Tastatur ist so konzipiert, daß beim Drücken einer Taste der entsprechende Tastaturcode auf dem Datenbus zur Verfügung gestellt wird. Dabei muß natürlich der Tastaturcode identisch mit dem auch im Bildschirminterface verwendeten Zeichencode gemäß ISO-7-Bit bzw. ASCII sein. Alle gewünschten Tastaturcodes werden auf einem PROM (*U 555*) gespeichert. Das Tastaturinterface sorgt nun dafür, daß beim Drücken einer Taste dem PROM gerade die Adresse bereitgestellt wird, auf der der dazugehörige Tastaturcode abgespeichert ist. Für die Meldung an das System über das Vorliegen eines gültigen Tastaturcodes steht ein Konsolstatussignal zur Verfügung.

Mit Hilfe eines Fertigmeldeimpulses kann man auch die softwaremäßige Übernahme des gültigen Tastaturcodes durch die CPU z. B. über nicht maskierbaren Interrupt ($\overline{\text{NMI}}$) einleiten.

Funktionsweise (Bild 6.1):

Ein Taktgenerator ($f \approx 1,5 \text{ KHz}$) steuert den 4-Bit-Binärzähler *D 193* an, dessen Ausgänge dem 1. Schaltkreis *D 195* (*D7*) eines insgesamt 6-Bit-Adreßlatch zugeführt werden. Die Zählerzustände des *D 193* (*D6*) bilden damit nach Übernahme in den Latch die 4 niederwärtigen Adreßbit *A0* . . . *A3* für den Tastatur-PROM. Gleichzeitig wird aus ihnen mit Hilfe des 1-aus-16-Decoders *MH 74154* das Spaltenauswahlsignal für die Kontaktmatrix der Tastatur gewonnen. Die Decoderausgänge werden dabei durch die Dioden *VD1* . . . *VD16* geschützt.



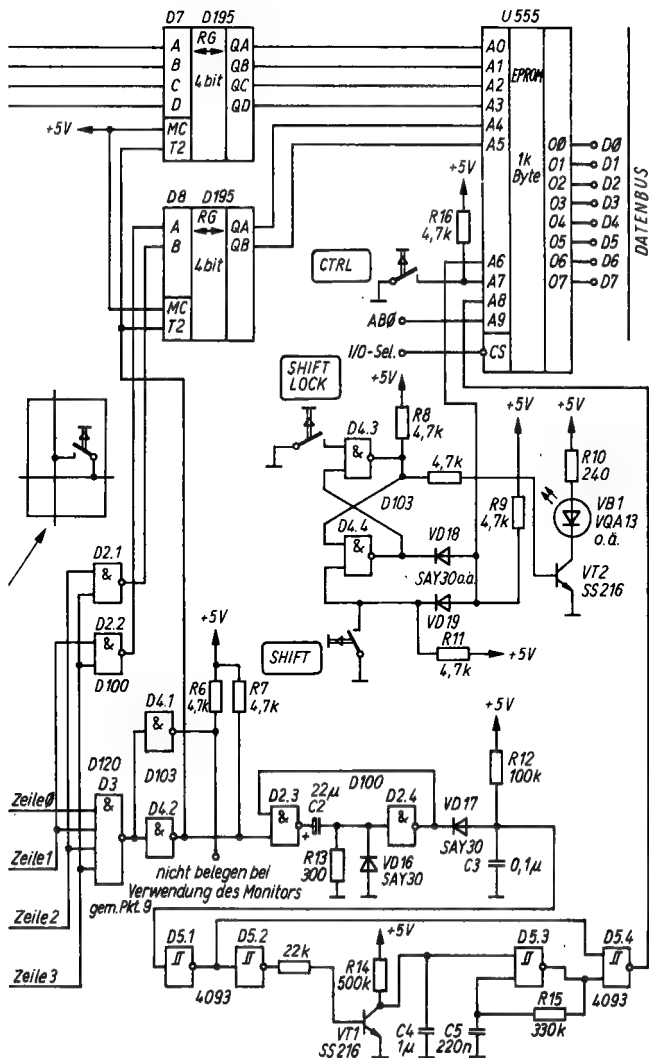


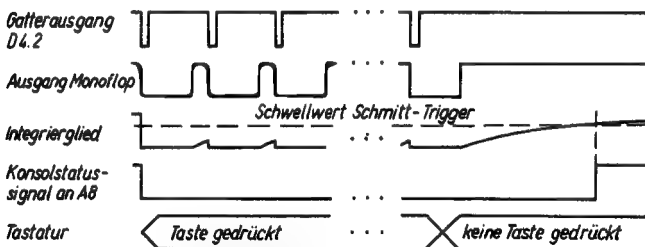
Bild 6.1 Alphanumerische Tastatur

Für den Fall, daß keine Taste gedrückt ist, liegen alle Zeilenpotentiale der Kontaktmatrix über den Widerständen R2 . . . R5 auf H-Pegel. Beim Schließen eines Tastenkontaktes in der Zeile 0 wird über den jeweiligen Decoderausgang das Zeilenpotential kurzzeitig auf L gezogen. An den Gattern D3, D4.2 entsteht ein Übernahmeimpuls für die 2 Schaltkreise *D 195* des Adreßlatch (HL – Flanke an T2). Damit ist im 1. *D 195* (D7) der gerade gedrückten Taste entsprechende Zählerzustand des *D 193* gespeichert und liegt als Adresse an den Eingängen A0 . . . A3 des Tastatur-PROM vor. Die adressierte Speicherzelle stellt den gewünschten Tastaturcode am Datenbus zur Verfügung.

Der Latchimpuls kann am Ausgang von Gatter D4.1 als Fertigimpuls abgenommen werden. Gleichzeitig steuert er einen Monoflop (bestehend aus den Gattern D2.3, 2.4) an, der das Konsolstatussignal erzeugt. Dieses Signal wird nach Durchlaufen eines Integrierglieds über die Gatter D5.1, 5.4 des Schmitt-Trigger-Schaltkreises 4093 dem Adreßeingang A8 des PROM zugeführt. Dazu liegt im Normalfall der Ausgang von Gatter D5.3 auf H. Wenn eine Taste länger, als bei normaler Betätigung üblich, gedrückt wird ($t = 0,5$ s), erreicht die Spannung an C4 den Schwellwert des Schmitt-Triggers D5.3, der als Generator geschaltet ist. Der Generator schwingt mit etwa 15 Hz, und das Konsolstatussignal steht an A8 in ständiger Wiederholung bereit. Damit wiederholt sich auch die Übernahme des Tastaturcodes so lange, wie die Tastatur gedrückt ist.

Kontaktprellungen machen sich durch den Ausfall eines oder mehrerer Impulse am Gatter D4.2 bemerkbar. Wie aus dem Impulsver-

Bild 6.2 Impulsdiagramm Konsolstatussignalerzeugung



Darstellung vereinfacht, nicht maßstäblich

lauf in Bild 6.2 zu ersehen ist, wählt man die Zeitkonstante des Integriergliedes deshalb so, daß im genannten Fall der Schwellwert des nachfolgenden Schmitt-Triggers nicht erreicht wird. Damit ist eine ausreichende Prellunterdrückung gewährleistet.

Beim Betätigen einer Taste in den Zeilen 1 . . . 3 laufen prinzipiell die gleichen Vorgänge ab. Zusätzlich werden jedoch die der höheren Tastenwertigkeit entsprechenden Adreßbit A4 und A5 durch die Gatter D2.1, 2.2 erzeugt und im 2. D 195 (D8) des Adreßlatch gespeichert.

Tabelle 6.1. enthält die sich ergebende Zuordnung der Tastenwertigkeiten, die der Bitbelegung A0 . . . A5 am PROM entspricht.

Das RS-Flip-Flop (D4.3, D4.4) ermöglicht die Umschaltung SHIFT bzw. SHIFTLOCK. Damit wird die bei Tastaturen übliche Zeitfunktion der Tasten (z. B. Groß- und Kleinbuchstaben) über den Adreßeingang A6 des Tastatur-PROM realisiert. Der Zustand SHIFTLOCK wird mit der LED VB1 angezeigt.

Mit der an den Adreßeingang A7 angeschlossenen CTRL-Taste lassen sich weitere Tastenfunktionen erzeugen, natürlich nur bei entsprechenden programmierten PROM.

Der Adreßeingang A9 wird mit der Adreßlinie AB0 des Systembusses verbunden und erfüllt eine Funktion im Zusammenhang mit der Konsolstatusabfrage.

Über den CS-Eingang wird die I/O-Portzuordnung der Tastatur realisiert. Dementsprechend verbindet man einen Ausgang des I/O-Portdecoders (in diesem Fall I/O-Port 08H) auf der CPU-Platine.

Prinzipiell lassen sich die Tasten in der Tastaturmatrix beliebig zuordnen. Es wird jedoch gefordert, daß auf der jeweils erzeugten PROM-Adresse der richtige Tastaturcode abgespeichert ist.

Tabelle 6.1. Zuordnung
der Tastenwertigkeiten

Zeiler der Kontaktmatrix	Tastenwertigkeit (A0 . . . A5)
0	0 . . . 15
1	16 . . . 31
2	32 . . . 47
3	48 . . . 63

Für die in Bild 6.1 gewählte Tastenzuordnung muß man den Tastatur-PROM *U 555* gemäß ISO-7-Bit-bzw. ASCII-Code entsprechend Tabelle 6.2. programmieren. In Tabelle 6.2. wurden neben den eigentlichen Tastaturcodes auch die zur Realisierung der Konsolstatusfrage notwendigen Randbedingungen berücksichtigt.

6.2. Aufbau und Inbetriebnahme

Der Aufbau einer alphanumerischen Tastatur erfordert eine sinnvolle, d. h. bedienungsgerechte, Tastenanordnung (ähnlich einer Schreibmaschinentastatur).

Bild 6.3 zeigt eine derartige Tastenanordnung, zugeschnitten auf das vorgestellte Mikrocomputersystem. Als Tasten eignen sich beispielsweise die im Amateurhandel erhältlichen Mikrotaster des VEB *Elektroschaltgerätekwerk Auerbach*.

Die Tastatur wird mit einem Kabel an den Computer angeschlossen. Bringt man den Tastatur-PROM unmittelbar in der Tastatur unter, dann ergibt sich damit eine unerwünschte Verlängerung des Systembusses. Deshalb ist es sinnvoll, den Tastatur-PROM im Computereinschub, d. h. in Systembusnähe, zu platzieren.

Zur Inbetriebnahme – vorerst ohne PROM-Bestückung – wird zunächst die Taktfrequenz von etwa 1,5 kHz am Ausgang von Gatter D1.2 bzw. D1.4 mit dem Oszilloskop kontrolliert. Anschließend führt man eine oszillographische Signalkontrolle an folgenden Stellen der Schaltung durch:

- Zählerausgänge *D 193* (D6) einschließlich Eingänge *D 195* (D7), *MH 74154* (D9)
- Decoderausgänge *MH 74154*

Das Decoderausgangssignal muß bei gedrückter Taste auch auf der jeweiligen Zeile erscheinen.

- Gatterausgang D3, D4.1, D4.2, Monoflop (D2.3, D2.4) Repeaterschaltung (D5.2, D5.3), Adreßeingang A8 am Tastatur-PROM.

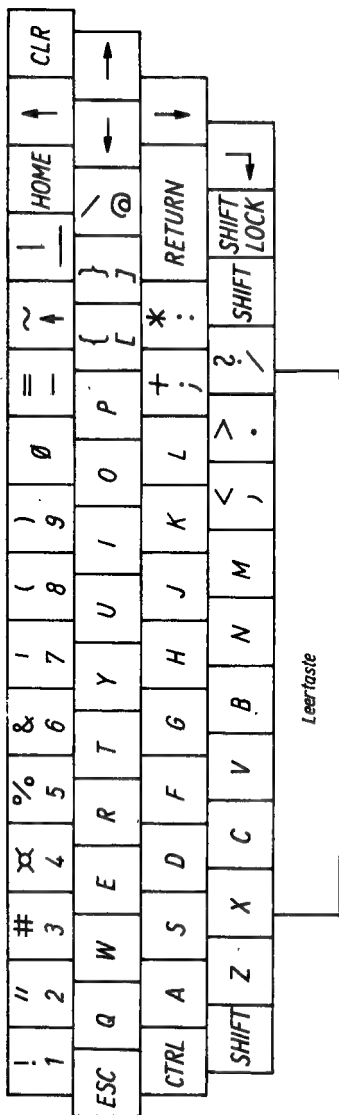
An diesen Punkten der Schaltung erscheint nur bei gedrückter Taste ein Signal. Damit schwingt also der Repeatgenerator, und am Gatterausgang D5.3 muß eine Rechteckspannung von etwa 15 Hz nachweisbar sein.

Sind alle Signale ordnungsgemäß vorhanden, steckt man den Tastatur-PROM auf und überprüft, ob nach dem Drücken einer Taste der richtige Tastaturcode an den Datenausgängen des PROM an-

Tabelle 6.2. HEXA-Tabelle Tastatur-PROM

Adresse HEXA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	01	02	03	04	05	06	07	08	09	00	1D	1E	1C	1B	11	17
010	05	12	14	19	15	09	0F	10	1B	1D	1C	01	13	04	06	07
020	08	0A	0B	0C	0B	0A	0D	10	18	03	16	02	0E	0D	1C	1E
030	1F	0C	0B	08	09	0A	1E	0E	20							
040	01	02	03	04	05	06	07	08	09	00	1D	1E	1C	1B	11	17
050	05	12	14	19	15	09	0F	10	1B	1D	1C	01	13	04	06	07
060	08	0A	0B	0C	0B	0A	0D	1A	18	03	16	02	0E	0D	1C	1E
070	1F	0C	0B	0C	09	0A	1E	05	20							
080	21	22	23	24	25	26	27	28	29	30	3D	7E	7C	1B	71	77
090	65	72	74	79	75	69	6F	70	7B	7D	5C	61	73	64	66	67
0A0	68	6A	6B	6C	2B	2A	0D	7A	78	63	76	62	6E	6D	3C	3E
0B0	3F	0C	0B	08	09	0A	1E	0E	20							
0C0	31	32	33	34	35	36	37	38	39	30	2D	5E	5F	1B	51	57
0D0	45	52	54	59	55	49	4F	50	5B	5D	40	41	53	44	46	47
0E0	48	4A	4B	4C	3B	3A	0D	5A	58	43	56	42	4E	4D	2C	2E
0F0	2F	0C	0B	08	09	0A	1E	0E	20							
100																
.																
.																
.																
1FF																
200																
.																
.																
.																
2FF																
300																
.																
.																
.																
3FF																

Anmerkung: Aus Gründen der Bedienungsfreundlichkeit des Monitorprogramms sind die Belegungen der Adreßbereiche 080H ... 0BFH und 0C0H ... 0FFH (= SHIFT/Normal) gegenüber ASCII bzw. ISO-7-Bit vertauscht.



Hinweis: Für die Tasten SHIFT, RETURN, Leertaste sollten möglichst 2 parallel geschaltete Mikro-taster verwendet werden.

Bild 6.3 Tastenanordnung für alphanumerische Tastatur

liegt. Gemäß ISO-7-Bit bzw. ASCII (siehe Anhang) muß sich z. B. für den Buchstaben A der Code 41 H ergeben, wenn nicht gleichzeitig die Taste SHIFT gedrückt ist (bzw. SHIFTLOCK nicht vorher gedrückt wurde). In dieser Art lassen sich alle Tastenfunktionen bzw. -codes kontrollieren.

Als praktisches Hilfsmittel für diese Kontrolle erweist sich die Verwendung von 8 LED mit entsprechenden Treibertransistoren. Notfalls erfüllt natürlich auch ein einfacher Vielfachmesser den gewünschten Zweck.

Das Zusammenspiel der Tastatur mit dem Computer kann mit einem kleinen Testprogramm bzw. dem in Abschnitt 9. behandelten Monitorprogramm überprüft werden.

Vor dem Start des im folgenden angegebenen Testprogramms sind der CS-Eingang des Tastatur-PROM auf das I/O-Port 08H und der Adreßeingang A9 auf die Adreßlinie AB0 des Systembusses zu legen. Der Bildschirm muß auf E800H selektiert sein.

```
M1:    LD HL, E7FFH
M2:    INC HL
        LD A, H
        CMP EEH
        JRZ M1
M3:    IN A, (09H)
        OR A
        JRNZ M3
M4:    IN A, (09H)
        OR A
        JRZ M4
        IN A, (08H)
        LD (HL), A
        JR M2
```

Bei richtiger Funktion der Tastatur werden die der jeweils gedrückten Taste entsprechenden Zeichen nacheinander, links oben beginnend, auf den Bildschirm geschrieben. Bei längerem Drücken einer Taste wirkt die Repeatfunktion. Solange die Taste gedrückt ist, wird ein Zeichen in ständiger Wiederholung ausgegeben.

7. Ein-/Ausgabebaugruppen

Der Computer kann bisher nur über die Tastatur und den Bildschirm Kontakt mit der Außenwelt aufnehmen. Mit geeigneten Ein-/Ausgabeschaltungen wird der Computer zu einem universell verwendbaren Gerät, mit dem sich auch die vielfältigsten Meß-, Steuer- und Regelaufgaben sowie eine Datenübertragung realisieren lassen. Einige Interfacetechniken sollen in diesem Abschnitt beschrieben werden.

7.1. Universelle I/O-Karte

Das Herz vieler Ein-/Ausgabeschaltungen ist eine universelle I/O-Leiterkarte. Die Leiterkarte enthält 2 PIO-Bausteine, 1 SIO-Baustein und 1 CTC-Baustein. Auf der Leiterkarte sind zusätzlich noch ein Kassetteninterface (siehe Abschnitt 8.), ein Port zum Abschalten des Computers und ein Tongenerator für Signalzwecke untergebracht.

Bild 7.1 und Bild 7.2 zeigen die Schaltung der Leiterkarte. Will man das Monitorprogramm aus Abschnitt 9. verwenden, dann müssen die IOSEL-Leitungen so mit dem zentralen I/O-Decoder verbunden werden, daß sich folgende Adressen ergeben (siehe auch Tabelle 3.2.):

00H Power-off-Port

04H Tongenerator aus

05H Tongenerator an

08H Daten Tastatur

09H Status Tastatur

0CH Daten PIO 1 Port A

0DH Steuerwort PIO 1 Port A

0EH Daten PIO 1 Port B

0FH Steuerwort PIO 1 Port B

10H Daten PIO 2 Port A

11H Steuerwort PIO 2 Port A

12H Daten PIO 2 Port B

13H Steuerwort PIO 2 Port B

14H Kanal 0	CTC
15H Kanal 1	CTC
16H Kanal 2	CTC
17H Kanal 3	CTC
18H Daten	SIO Port A
19H Steuerwort	SIO Port A
20H Daten	SIO Port B
21H Steuerwort	SIO Port B

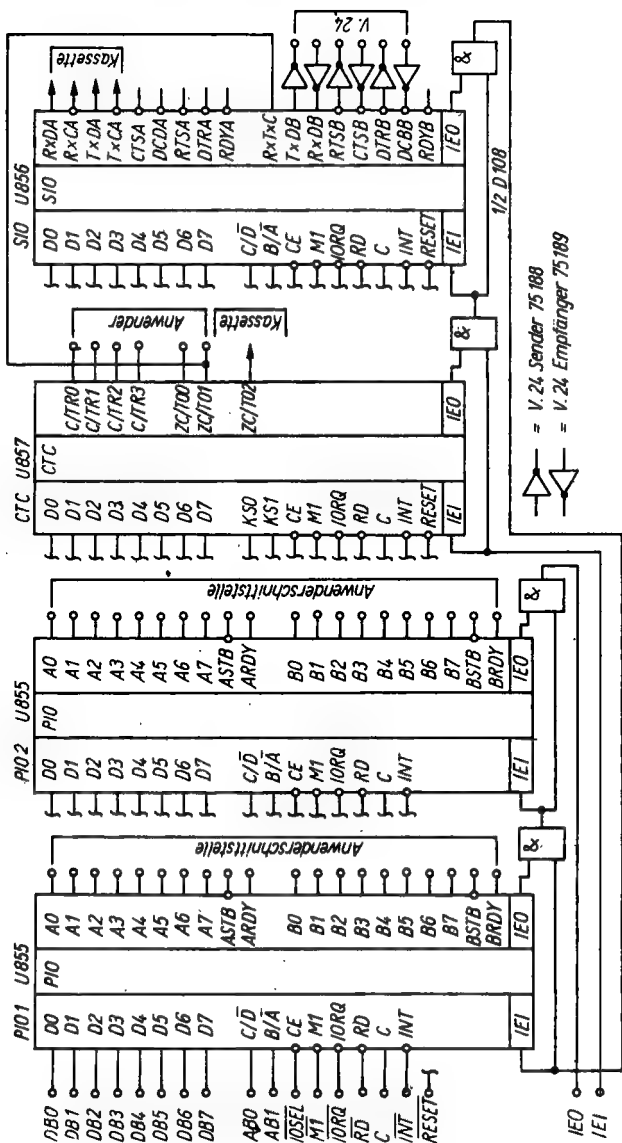
Die PIO 1 wird im System zum Anschluß eines Lochbandlesers genutzt, Port A überträgt die Daten, Port B die Steuersignale. Mit der PIO 2 stehen weitere 16 Ein-/Ausgabeleitungen zur Verfügung.

Zur seriellen Ein-/Ausgabe ist eine SIO vorgesehen. Die SIO wandelt 8-Bit-parallele in 1-Bit-serielle Informationen um. Somit sind nur sehr wenige Verbindungsleitungen zur Datenübertragung notwendig. Weiterhin arbeiten viele externe Geräte von Natur aus seriell, beispielsweise Fernschreiber, Floppys und Magnetbandspeicher. Ein Interface zur Datenaufzeichnung auf Magnetband wird in Abschnitt 8. beschrieben. Die Datenaufzeichnung auf Magnetband ermöglicht das SIO Port A.

Zum seriellen Informationsaustausch zwischen dem Computer und anderen Geräten ist eine V24-Schnittstelle vorgesehen (SIO Port B). Anpaßstufen sorgen für die Erzeugung der notwendigen Spannungspegel. Die V24-Schnittstelle arbeitet mit den Pegeln +12 V (0-Bit) und -12 V (1-Bit). Durch die Wahl dieser Pegel wird eine sichere Datenübertragung auch über größere Entfernungen garantiert. Außer den seriellen Sende- und Empfangsdaten (TxD und RxD) werden noch folgende Steuersignale auf die Schnittstelle geführt:

RTSB Sendeaufforderung	(Ausgang)
DTRB Datenstation bereit	(Ausgang)
DCDB Empfängerfreigabe	(Eingang)
CTSB Senderfreigabe	(Eingang)

Mit diesen Steuersignalen können verschieden schnelle Geräte miteinander synchronisiert werden. Beispielsweise läßt sich mit Hilfe des CTS-Eingangs das Senden von Daten verhindern, wenn das periphere Gerät die eingetroffenen Daten noch nicht verarbeitet hat. Umgekehrt kann das Signal RTS verhindern, daß das periphere Gerät weitere Daten sendet, solange der Computer die empfangenen Informationen noch nicht verarbeitet hat. Dazu muß



2 D108

Bild 7.1 Ein-/Ausgabeplatine

man die Anschlußpunkte CTS und RTS des Computers und des peripheren Geräts kreuzweise miteinander verbinden. Die Daten werden mit genormten Geschwindigkeiten übertragen.

Üblich sind z. B. folgende Übertragungsraten (Baudraten, 1 baud = 1 Bit/s):

45, 45, 50, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200.

Die Sende- und Empfangstakte erzeugt ein CTC-Baustein. Durch entsprechende Programmierung des CTC können die verschiedenen Baudraten eingestellt werden (siehe auch Abschnitt 8.). Der CTC-Kanal 2 ist mit dem Kassetteninterface und der CTC-Kanal 1 mit dem Sende-/Empfangstakteingang RxTxCB des SIO-Schaltkreises verbunden. Die anderen beiden CTC-Kanäle stehen dem Anwender frei zur Verfügung.

Bei systemseitigem Ansteuern der I/O-Leiterkarte wurde auf den Einsatz von Treiberschaltkreisen verzichtet. Beim Aufbau eines sehr großen Systems wird jedoch der Einsatz von Treiberschaltkreisen (z. B. 8216) empfohlen. Die AND-Gatter verringern die Einschwingzeiten der Interruptprioritätskette [2].

Auf der I/O-Karte fanden noch 2 kleine Schaltungen Platz, die in Bild 7.2 zu sehen sind. Die Schaltkreise D1, D2, D4.3 und N1 bilden einen Tongenerator, der mit dem Befehl OUT 5 angeschaltet und mit OUT 4 wieder abgeschaltet werden kann. Das JK-Flip-Flop D2 arbeitet als Frequenzteiler und als Tor. Das Rechtecksignal wird vom Transistor VT1 verstärkt und dem (hochohmigen) Lautsprecher zugeführt.

Wenn wie im Mustergerät die Netzzuschaltung mit einer Relais-selbsthalteschaltung erfolgt, so kann mit der 2. Schaltung der Computer softwaregesteuert abgeschaltet werden. Der Kontakt des Relais K unterbricht nach der Befehlssequenz OUT 0, OUT 0 und nach einer Zeit von etwa 18 s die Spannungszuführung des Netzrelais. Die Zeitverzögerung wird vom RC-Glied am Eingang des Triggerschaltkreises N2 bestimmt. Das Schieberegister D3.1 und D3.2 verzögert das Aufladen des Kondensators um einen OUT-Befehl. Mit IN0-Befehlen, RESET oder NMI kann die Abschaltlogik zurückgestellt werden. Der Transistor VT2 beschleunigt in diesen Fällen das Entladen des Kondensators.

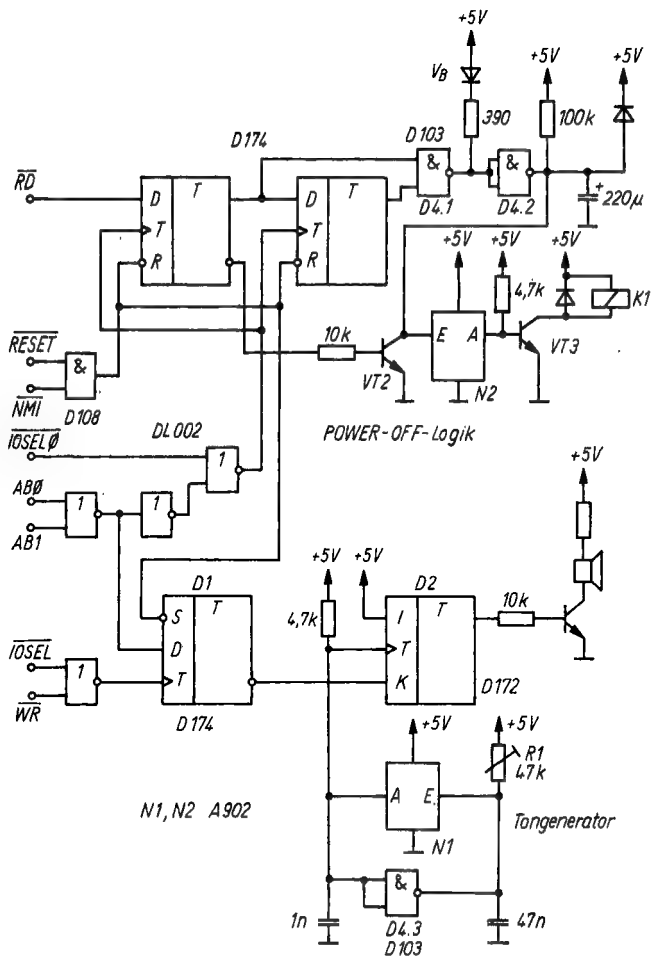


Bild 7.2 Power-off-Schaltung, Tongenerator

7.2. EPROM-Programmiergerät

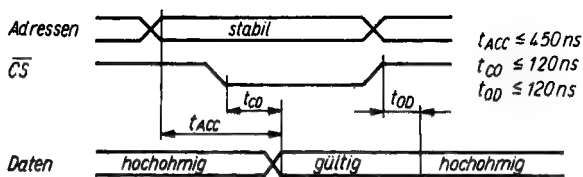
Ein wichtiger Zusatz zum Computer ist das EPROM-Programmiergerät. Da man zukünftig auch höher integrierte EPROM-Schaltkreise erhalten wird, wurde es so entwickelt, daß sich nicht nur Speicherschaltkreise vom Typ *U555* bzw. *2708* programmieren lassen, sondern das gesamte Typenspektrum: vom „Kleinsten“ *2704* bis zum *27256*. Die Anschlußbelegungen der verschiedenen EPROM-Typen sind im Anhang enthalten. Ein Vergleich der Pin-Belegungen zeigt, daß es nur wenige Unterschiede gibt. Bei EPROM-Schaltkreisen im 24poligen DIL-Gehäuse sind es die Pins 18 ... 21 und 24. Über diese Pins werden \overline{CS} , Programmierspannung und Programmierimpulse, Betriebsspannungen und/oder weitere Adreßbit zugeführt. EPROM-Schaltkreise ab *2764* befinden sich im 28poligen Gehäuse. Bei diesen Schaltkreisen kommen noch 4 Anschlüsse hinzu, für die gleiches gilt. Die genannten Signale werden über Codierstecker zugeführt. Mit Hilfe dieser Codierstecker wird die Hardware an den jeweiligen EPROM-Typ angepaßt.

7.2.1. Programmiervorschriften

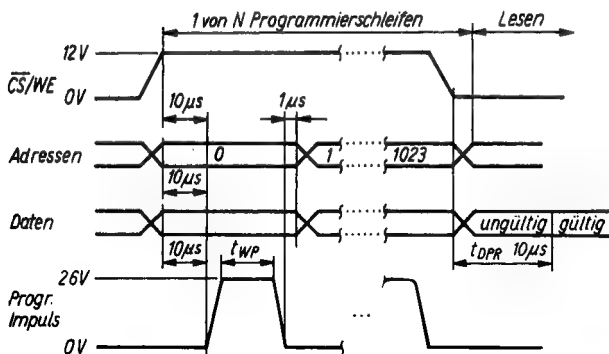
Im unprogrammierten Zustand des EPROM-Schaltkreises sind alle Bit 1 (Ausgänge H-Pegel). Programmiert wird somit durch Einschreiben von 0 Bit in die adressierten Zellen. Eine programmierte 0 kann nicht mehr umprogrammiert werden.

Bild 7.3 verdeutlicht das Programmieren des *U555* an Hand des Impulsdigramms. Den \overline{CS}/WE -Eingang legt man auf +12 V, anschließend die 1. Adresse und das zugehörige Datenwort (Ausgänge $0_0 \dots 0_7$) an den EPROM. Es folgt ein Programmierimpuls (0,1 ... 1 ms, 26 V). Danach wird die Adresse um 1 erhöht und der Vorgang wiederholt, bis alle Adressen abgearbeitet sind. Das muß N-mal für alle 1024 Adressen wiederholt werden. Die Anzahl der Programmierschleifen N hängt von der Breite des Programmierimpulses t_{pw} ab. Die integrale Programmierdauer je Speicherplatz beträgt 100 ms, so daß bei $t_{wp} = 1$ ms 100 Programmierschleifen notwendig sind (*Intel*-Programmiervorschrift).

Moderne EPROM lassen sich einfacher programmieren. Der Schaltkreis *2716* wird mit 50-ms-TTL-Impulsen programmiert. Er



a)



b)

Bild 7.3 a – Impulsdiagramm U555, b – Impulsdiagramm 2708

gestattet das Programmieren einzelner Speicherzellen und das Prüfllesen während des Programmiervorgangs. Das Programmieren kann folgendermaßen ablaufen: Anlegen der Programmierspannung U_{pp} (25 V) an Pin 21, $\overline{\text{OE}} = \text{H-Pegel}$, Anlegen Adresse und Datenbyte, Programmierimpuls (max. 55 ms) dem Eingang $\overline{\text{CE}}$ zuführen. Ein Prüfllesen ist möglich, wenn der $\overline{\text{OE}}$ -Eingang auf L gelegt wird, dabei muß U_{pp} nicht abgeschaltet werden. Bild 7.4 zeigt das Impulsdiagramm.

Der Schaltkreis 2732 wird ähnlich programmiert (Bild 7.5). Die Programmierspannung (25 V) wird dem Eingang $\overline{\text{OE}}/V_{\text{pp}}$ zugeführt, der Programmierimpuls (50 ms; L-Pegel) dem Eingang $\overline{\text{CE}}$.

Die genauen Programmiervorschriften sind unbedingt den Datenblättern des Herstellers zu entnehmen. Beispielsweise wird für den U 555 eine integrale Programmierzeit von 50 ms je Speicherzelle

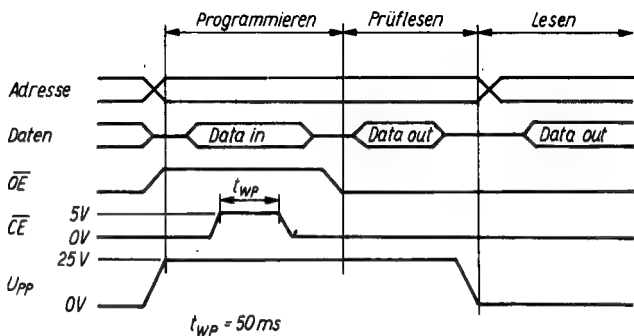


Bild 7.4 Impulsdiagramm 2716

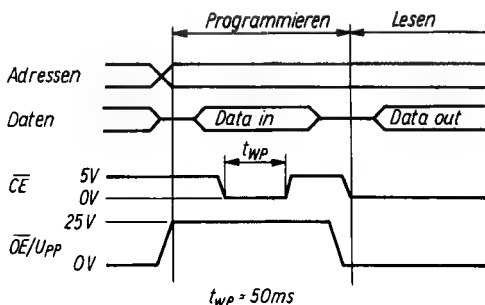


Bild 7.5 Impulsdiagramm 2732

[6], für den *Intel 2708* aber 100 ms [7] angegeben. Ein weiterer Hinweis in diesem Zusammenhang: Die EPROM-Schaltkreise *TMS 2716* und *i 2716* sind nicht kompatibel.

7.2.2. Die Schaltung

Bild 7.6 zeigt die Schaltung des Programmiergeräts. Ein PIO-Baustein überträgt die Daten und Steuersignale von und zum Programmiergerät. Die Daten werden über Port A übertragen. Um einen PIO-Baustein einzusparen, erzeugt man die Adressen hardwaremäßig. Dafür sind die Zähler D1 ... D4 vorgesehen. Der Zähler

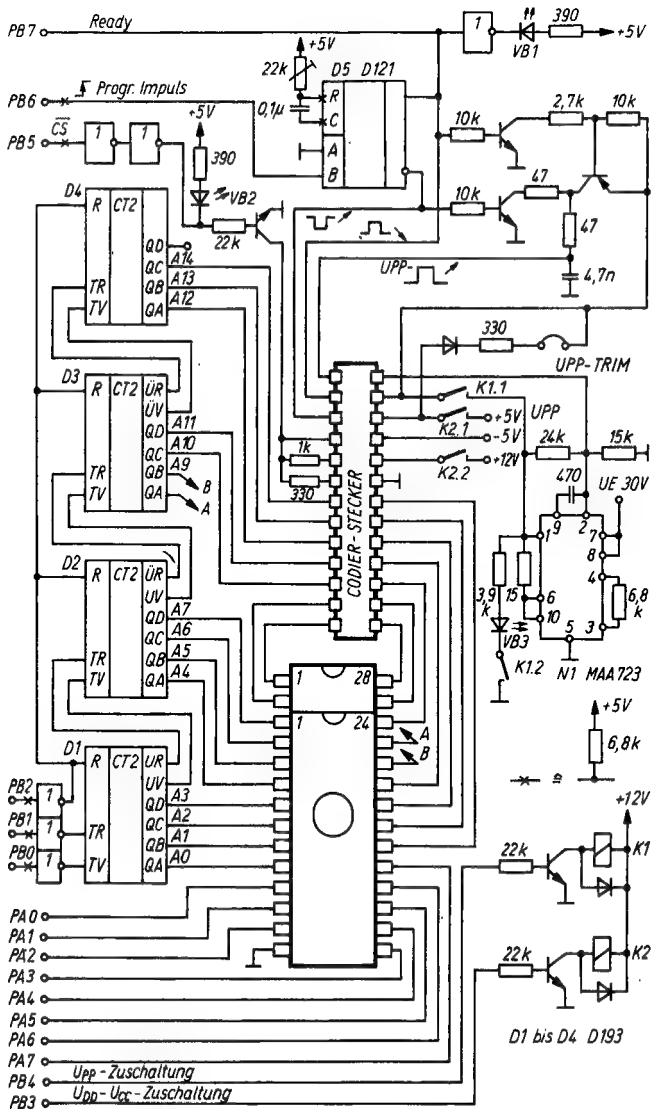


Bild 7.6 EPROM-Programmiergerät

D4 wird erst notwendig, wenn auch EPROM-Schaltkreise mit einer Kapazität > 4 kByte programmiert werden sollen. Die Zählerausgänge sind mit dem EPROM-Sockel (A0 bis A9) und dem Codierstecker (A10 bis A14) verbunden.

Den Programmierimpuls von 1 ms Länge erzeugt der Monoflop D5. Diese Variante bietet gegenüber einer softwaremäßigen Lösung den Vorteil, daß auch im Havariefall die Impulsbreite 1 ms nicht überschritten und so ein Zerstören des EPROM verhindert wird. Um bei der Programmierung der Typen 2716, 2732, 2764 usw. die notwendige Programmierzeit von 50 ms je Speicherplatz zu erreichen, wird der monostabile Multivibrator 50mal angestoßen. Die Ausgänge Q und \overline{Q} des Monoflop werden auf den Codierstecker geführt. Diese beiden Signale nutzt man bei den Schaltkreisen 2716, 2732 usw. als Programmierimpuls. Beim Programmieren der Typen U 555, 2708, TMS 2716 muß der 1-ms-Impuls auf etwa 26 V verstärkt werden. Dafür ist der Verstärker mit den Transistoren VT1 . . . VT3 vorgesehen. Die 26-V-Impulse gelangen ebenfalls an den Codierstecker. Programmierimpulse zeigt die Leuchtdiode VB1 an.

Der Transistor VT4 arbeitet als Schaltstufe für das \overline{CS}/WE -Signal beim U 555 bzw. 2708 oder für das \overline{OE} -Signal beim 2716 und 2732. Den Zustand dieses Signals zeigt die Leuchtdiode VB2 an. Mit dem Relais K1 wird die Programmierspannung zugeschaltet. Die Leuchtdiode VB3 hat Kontrollfunktion. Das Programmiergerät enthält ein Netzteil für die Programmierspannung U_{pp} . Die Spannung stabilisiert man mit dem Schaltkreis N1 (MAA 723). Der Knotenpunkt der Widerstände R1/R2 wird auf den Codierstecker geführt, um dort die genaue Programmierspannung mit einem Zusatzwiderstand festzulegen. Das Relais K2 schaltet die Betriebsspannung des EPROM-Schaltkreises softwaregesteuert zu.

7.2.3. Aufbau und Codierung

Die Programmiereinrichtung wurde als externes Gerät aufgebaut und läßt sich bei Bedarf über ein Kabel mit der PIO 2 verbinden. Das hat den Vorteil, daß die PIO 2 auch für andere Ein-/Ausgabearbeiten genutzt werden kann.

Als EPROM-Fassung sollte ein 28poliger Nullkraftsockel eingesetzt werden. Ein 26poliger direkter Steckverbinder nimmt die Co-

dierstecker auf. Sie bestehen aus 30 mm × 35 mm großen doppelt kaschierten Leiterplatten. Bild 7.7 zeigt die Anschlußbelegung der Codiereinrichtung. Bild 7.8 bis Bild 7.10 zeigen die Beschaltung der Pins 18 . . . 21 sowie die Verdrahtung der Codierstecker für die EPROM-Typen *U 555 (2708)*, *2716* und *2732* (vergleiche auch Bild 7.3 bis Bild 7.5).

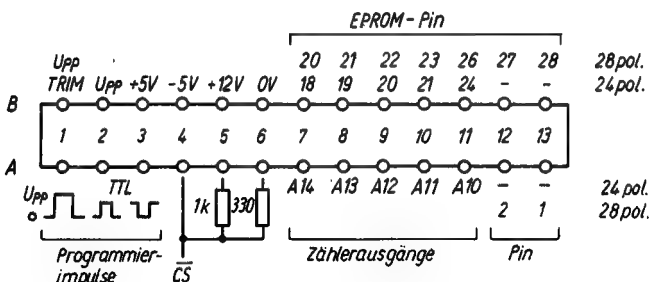


Bild 7.7 Anschlußbelegung der Codiereinrichtung

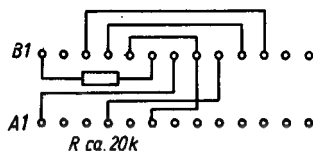


Bild 7.8 Codierstecker für *U 555 (2708)*

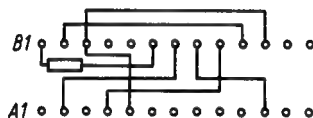


Bild 7.9 Codierstecker für *2716*

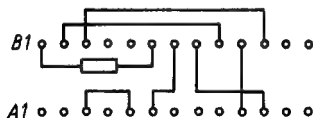


Bild 7.10 Codierstecker für *2732*

Die Verbindung des Programmiergeräts mit dem Computer übernimmt die PIO 2. Dabei sind die beiden Ports wie folgt belegt:

Port A Daten

Port B 0 Vorwärtszählen (positiver Impuls)
 1 Rückwärtszählen (positiver Impuls)
 2 Rücksetzen des Zählers (negativer Impuls)
 3 Zuschalten U_{∞} und U_{DD} (H)
 4 Zuschalten U_{pp} (H)
 5 $\overline{CS}/\overline{WE}$ bzw. \overline{OE} (H = Lesen)
 6 Programmierimpuls (L/H-Flanke)
 7 Fertigsignal (H für die Dauer des Impulses)

7.2.4. Ein Beispielprogramm

An einem BASIC-geschriebenen Programm soll das Programmieren eines U 555 demonstriert werden. Dabei wird abweichend von der Intel-Vorschrift programmiert [8], [9].

```
10 PRINT
20 PRINT „U 555/2708 PROGRAMMER“
30 PRINT
60 DA = 16:CA = 17:DB = 18:CB = 19:REM
   Ports festlegen
70 OUT CB, & CF:OUT CB, & 80:REM Bitmode, Bit
   7 Eingabe
80 OUT CA, & 7F:REM Byteeingabe
90 OUT DB, & 21:REM Spannungen aus, Zählerreset.
100 INPUT „LESEN (1) PROG (2) VERGLEICHEN (3)
   LÖSCHTEST (4)“; M
110 IF M > 4 THEN END
120 ON M GOSUB 250, 320, 550, 140
130 GOTO 90
140 REM Löschkontrolle
150 OUT CA, & 7F:REM Byteeingabe
160 OUT DB, & 29:OUT DB, & 2D:REM
170 F = 0:REM Flag
180 FOR I = 1 TO 1024:IF INP (DA) <> & FF THEN F =
   1:I = 1024
190 OUT DB, & 2c:OUT DB, & 2D:REM weiterzählen
200 NEXT I
```

```

210 PRINT „EPROM“; : IFF = 1 THEN PRINT „NICHT“;
220 PRINT „GELÖSCHT“ : RETURN
250 REM Dublizieren des EPROM-Inhalts ins RAM
260 INPUT „ADRESSE“; A
270 OUT CA, & 7F : REM Byteeingabe
280 OUT DB, & 29 : OUT DB, & 2D : REM Reset
290 FOR I = A TO A + 1023 : POKE I, INP (DA)
300 OUT DB, & 2c : OUT DB, & 2D : REM weiterzählen
310 NEXT I : RETURN
320 REM programmieren
330 INPUT „ADRESSE“; A : GOSUB 150 : IFF = 1 THEN
INPUT „WEITER (J/N)“; W$ : IF W$ <> „J“ THEN
RETURN
340 PRINT : Z = 0 : REM Zykluszähler = 0
350 N = 2 : GOSUB 430 : REM 2 Programmiersversuche
360 REM Prüfllesen
370 OUT CA & 7F : OUT DB, & 2D : OUT DB, & 29 : OUT
DB, & 2D
380 F = 0 : FOR I = A TO A + 1023 : IF INP (DA) <> PEEK
(I) THEN F = 1 : I = A + 1024
390 OUT DB, & 2C : OUT DB, & 2D : NEXT I
400 IF F = 1 AND Z < 32 THEN 350
410 IF F = 1 THEN PRINT „NICHT PROGRAMMIERT“;;
GOTO 520
420 PRINT : N = Z/2 : GOSUB 430 : GOTO 520 : REM
Sicherheitszyklen
430 OUT DB, & 29 : OUT DB, & 19 : OUT DB, & 1D : REM
Zählerreset und Spannungen anlegen
440 OUT CA, & F : REM Byteausgabe
450 FOR I = 1 TO N : Z = Z + 1 : PRINT CHR$ (&B);
USING „PROG ZYKLUS # # #“; Z
460 OUT DB, & 19 : OUT DB, & 1D : REM Zähler
rücksetzen
470 FOR I = A TO A + 1023 : OUT DA, PEEK (I)
475 REM Programmierimpuls erzeugen und weiterzählen
480 OUT DB, & 5D : OUT DB, & 1D : OUT DB, & 1C :
OUT DB, & 1D
490 NEXT I, J : RETURN
500 REM Auf Fertigsignal (Bit 7) wird nicht gewartet,
weil Schleifenlaufzeit > 1 ms

```



```

510  OUT CA, & 7F: REM Byteeingabe
520  OUT DB, & 2D: REM Auf Lesen schalten
525  REM Prüfllesen, anschließend Signalton
530  GOSUB 560: OUT 5,0: FOR I = 1 TO 500: NEXT
    I: OUT 4,0
540  RETURN
550  INPUT „ADRESSE“; A
560  OUT CA, & 7F: REM Byteeingabe
570  OUT DB, & 29: OUT DB, & 2D: REM Zähler
    rücksetzen
580  E = 0: FOR I = A TO A + 1023
590  IF INP (DA) <> PEEK (I) THEN E = E + 1
600  OUT DB, & 2C: OUT DB, & 2D: REM weiterzählen
610  NEXT I: PRINT E; „FEHLER“: RETURN

```

7.3. Analogschnittstellen

Will man mit dem Computer auch Analogsignale oder Meßwerte erfassen und verarbeiten, so sind geeignete Analog/Digital-Wandler und Digital/Analog-Wandler erforderlich. In diesem Abschnitt werden einige Analogschnittstellen beschrieben.

7.3.1. Digital/Analog-Wandler

International gibt es eine große Anzahl hybrider und monolithischer D/A-Wandlerbausteine. Ein besonders günstiger Baustein ist der sowjetische CMOS-Wandler *K 572 ИА1*. Es handelt sich bei diesem Schaltkreis um einen TTL-kompatiblen multiplizierenden 10-Bit-Wandler mit einer Einschwingzeit von 500 ns. Die Betriebsspannung kann 5 . . . 15 V ($P_{vmax} = 20$ mW), die Referenzspannung maximal ± 10 V betragen. Bild 7.11 zeigt das Funktionsprinzip des *K 572 ИА1* und Bild 7.12 die Standardbeschaltung. Die Ausgangsspannung beträgt

$$U_A = U_{ref} \sum_{i=1}^n \frac{S_i}{2n - i + 1}.$$

Dabei gilt: $S_i = 1$ bei H und $S_i = 0$ bei L am Digitaleingang E_i und $n = 10$.

Bild 7.13 zeigt die Beschaltung des D/A-Wandlers für bipolare

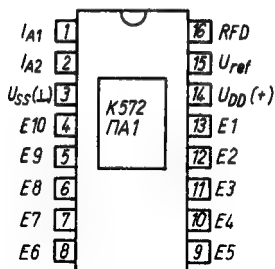
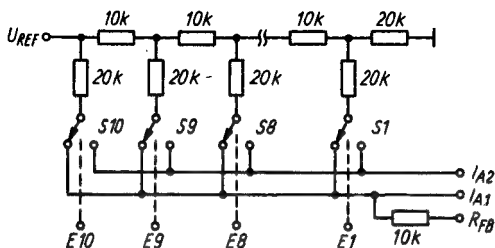


Bild 7.11 D/A-Wandler K 572 IIA1

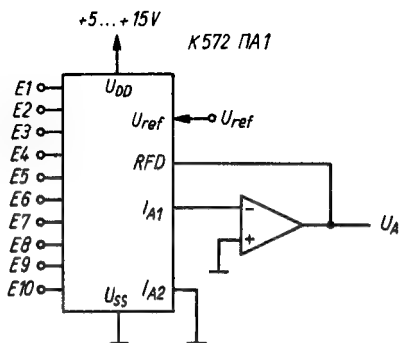


Bild 7.12
Standardbeschriftung
des K 572 IIA1 für unipolare
Ausgangsspannung

Ausgangsspannung. Die erreichbare Umsetzgeschwindigkeit ist von der Slew-Rate der verwendeten Operationsverstärker abhängig. Der A/D-Wandler K 572 IIA1 gestattet einfache und elegante Problemlösungen. Er läßt sich beispielsweise in den Gegenkopplungs-
zweig eines Verstärkers schalten, um eine digitale Verstär-

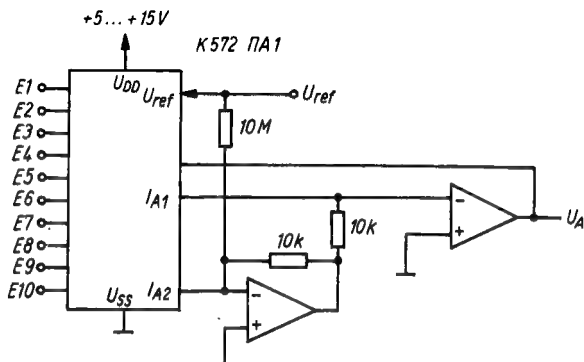


Bild 7.13 Standardbeschaltung des K 572 PA1 für bipolare Ausgangsspannung

kereinstellung zu ermöglichen, oder man verwendet ihn als digital steuerbaren Spannungsteiler. Dabei können von dem D/A-Wandler auch Wechselspannungen bis zu $U_{ss} = 20\text{ V}$ verarbeitet werden. Der Wandler läßt sich über einen PIO-Baustein an den Computer ankoppeln. Sind keine speziellen D/A-Wandlerschaltkreise verfügbar, so bieten sich auf der Grundlage von R-2R-Netzwerken verschiedene Lösungsvarianten an. Bild 7.14 und Bild 7.15. zeigen

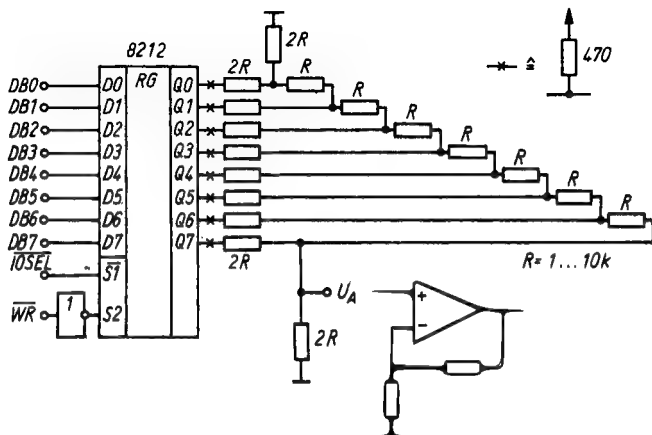


Bild 7.14 D/A-Wandler mit 8212

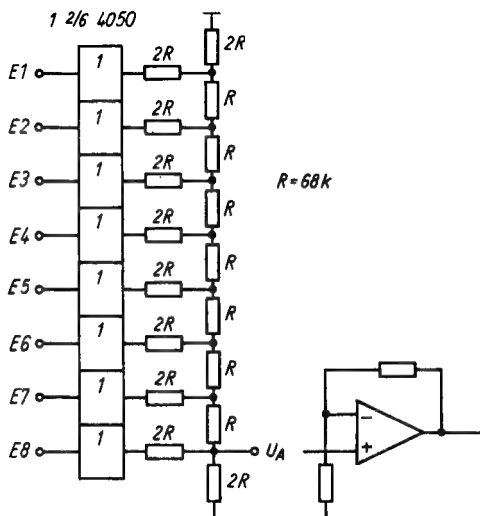


Bild 7.15 D/A-Wandler mit CMOS-Schaltkreisen 4050

2 einfache Schaltungen. Die Ausgangsstufen der Schaltkreise arbeiten als Spannungsschalter. Die Betriebsspannung der Schaltkreise ist gleichzeitig die Referenzspannung. Die Genauigkeit dieser D/A-Wandler hängt von der Toleranz der Widerstände und von der Charakteristik der Spannungsschalter ab. Die Durchlaßwiderstände der Schalter müssen klein gegenüber R , die Sperrwiderstände groß gegenüber R sein, damit die Fehlerströme im Netzwerk klein genug bleiben. Wenn keine hohen Anforderungen an die Genauigkeit gestellt werden, ist die Schaltung in Bild 7.14 für viele Anwendungen geeignet. Der Schaltkreis 8212 arbeitet als Port und als Spannungsschalter. Günstigere Eigenschaften hinsichtlich der Genauigkeit hat die Schaltung in Bild 7.15. Hier werden die Spannungsschalter durch CMOS-Leistungstreiber realisiert. Die Widerstände sollten Metallschichttypen mit einer Toleranz 1 % aus einer Fertigungscharge sein. Die Ausgangsspannung kann mit der bereits angegebenen Gleichung berechnet werden ($U_{Ref} = U_{cc}$, $n \triangleq$ Schaltung).

In Anwendungsfällen, bei denen es nicht so sehr auf hohe Umsetzgeschwindigkeiten ankommt, bieten sich D/A-Wandler an, die

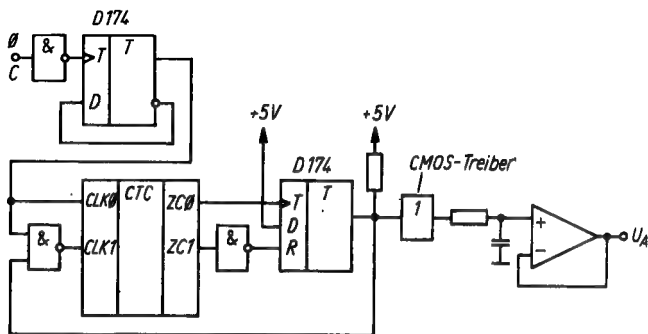


Bild 7.16 D/A-Wandler mit CTC-Baustein

durch die Pulsbreitenmodulation und anschließende Integration sehr genau umsetzen. Diese D/A-Wandler haben den Vorteil, daß sie sich ohne Präzisionsbauelemente aufbauen lassen [10]. Bild 7.16 zeigt einen 8-Bit-Digital/Analog-Umsetzer mit einem CTC-Baustein (siehe auch [2]). Beide CTC-Kanäle arbeiten im Zählermode. Der CTC-Kanal 0 erzeugt den Referenztakt ($TC = 256$). Der Kanal 1 wird mit dem zu wandelnden Wert geladen.

Mit 2 Digital/Analog-Wandlern (z. B. Bild 7.14) ist der Aufbau eines einfachen Oszillographen-Interface möglich. Bei dieser Anwendung wird der eine DAC-Ausgang mit dem X-Eingang, der andere DAC-Ausgang mit dem Y-Eingang des Oszillographen verbunden. Hat der Oszillograph einen Z-Eingang, so kann mit einem weiteren D/A-Wandler (geringerer Auflösung) auch eine Helligkeitsmodulation durchgeführt werden. Mit dieser Konfiguration ist eine einfache graphische Ausgabe möglich. Mit einem geeigneten Analog/Digital-Wandler läßt sich der Computer dann als NF-Speicheroszilloskop betreiben (bis einige hundert Hertz).

7.3.2. Analog/Digital-Wandler

Mit Hilfe der beschriebenen Wandler ist auf einfache Weise eine Analog/Digital-Wandlung möglich. Bild 7.17 und Bild 7.18 zeigen 2 mögliche Schaltungsvarianten, wobei die Schaltung in Bild 7.18 gleichzeitig 8 Analogkanäle erfassen kann. Das Signal wird soft-

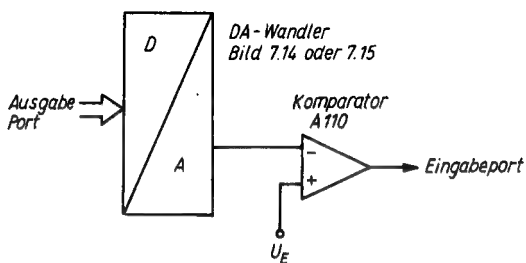


Bild 7.17 1-Kanal-A/D-Wandler

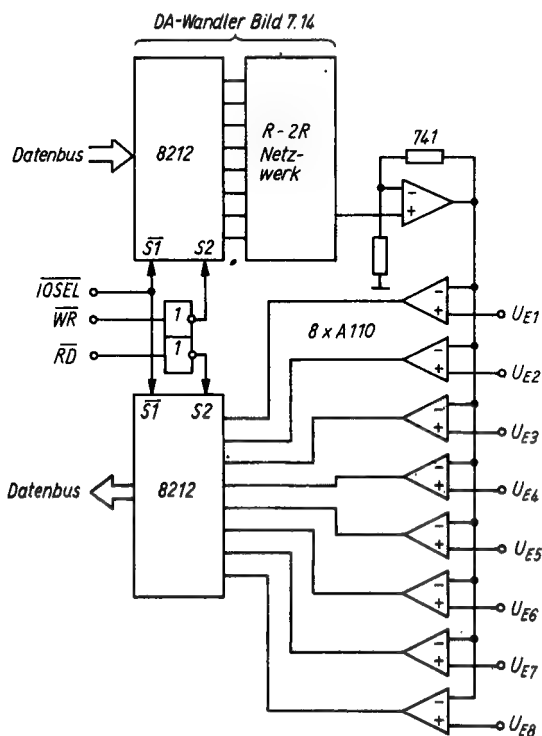


Bild 7.18 8-Kanal-A/D-Wandler

waregesteuert nach dem Verfahren der sukzessiven Approximation oder durch softwaremäßiges Nachbilden eines Vor-/Rückwärtszählers umgesetzt. Ein Komparator vergleicht die Ausgangsspannung des D/A-Wandlers mit der Eingangsspannung. Der Computer überwacht den Komparatorausgang. Je nach Pegelzustand des Komparators verändert die A/D-Software das Digitalwort, das anschließend an den D/A-Wandler ausgegeben wird. Ein Schaltungsbeispiel mit Vor- und Rückwärtszählern (Bild 7.19) soll verdeutlichen, wie man vorgeht. Die Ausgänge der Zähler D1 und D2 sind mit einem D/A-Wandler verbunden. Die Ausgangsspannung des D/A-Wandlers wird vom Komparator N1 mit der umzusetzenden Eingangsspannung verglichen. Das Komparatorausgangssignal, auf ein D-Flip-Flop geführt, bestimmt die Zählrichtung. Bei veränderter Eingangsspannung wird, je nach Vorzeichen der Änderung, vor- oder rückwärts gezählt, bis die Eingangsspannung und DAC-Ausgangsspannung gleich sind. Die Umsetzzeit ist

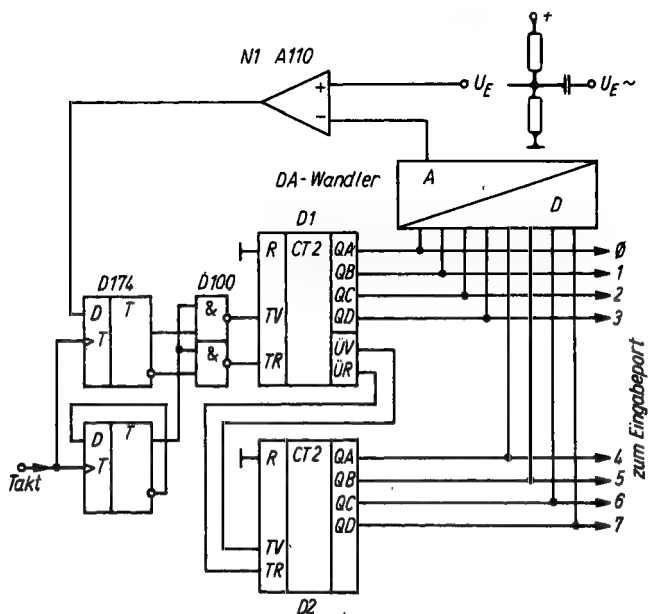


Bild 7.19 A/D-Wandler mit Vor-/Rückwärtszähler

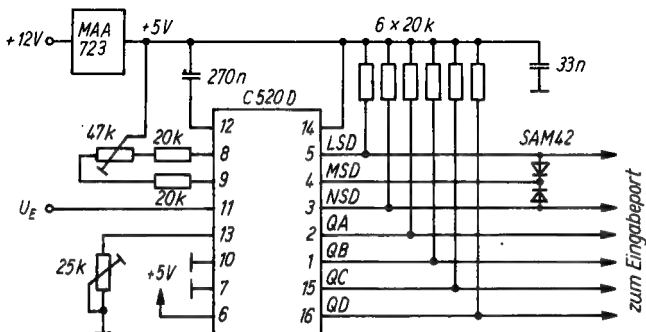


Bild 7.20 A/D-Wandler mit C 520

von der Größe der Änderung und der Taktfrequenz abhängig. Bei der Wahl der Taktfrequenz muß man die Einschwingzeiten des D/A-Wandlers und des Komparators berücksichtigen. An den Zählerausgängen kann das Digitalwort abgenommen und einem Eingabeport zugeführt werden. In vielen Fällen ist es zur Verringerung von Umsetzfehlern notwendig, dem Komparator eine Sample & Hold-Baugruppe vorzuschalten, die die Eingangsspannung während der Umsetzperiode konstant hält. Mit derartigen A/D-Wandlern lassen sich, vorausgesetzt, die Taktfrequenz kann genügend hoch gewählt werden, auch NF-Signale digitalisieren. Dann sind z. B. Anwendungen wie das oben erwähnte Speicheroszilloskop oder eine digitale Tonverarbeitung in der Musikelektronik möglich.

Für die Anwendungen, bei denen es nicht auf eine hohe Umsetzrate ankommt, bieten sich integrierte A/D-Wandler an. Bild 7.20 zeigt eine aufwandarme Schaltungsvariante. Die A/D-Umsetzung übernimmt der für Schalttafelinstrumente vorgesehene Schaltkreis C 520. In [11] wurde auch die notwendige Software veröffentlicht, so daß hier nicht auf weitere Einzelheiten eingegangen werden soll.

8. Kassetteninterface

Als externe Daten- und Programmspeicher setzt der Amateur vorwiegend Lochbänder und Magnetbänder ein. Das Magnetband ist ein preiswerter Datenspeicher hoher Kapazität. Im folgenden wird ein Kassetteninterface beschrieben, das die Aufzeichnung und die Wiedergabe von Daten und Programmen mit einem handelsüblichen Kassettenrecorder ermöglicht.

8.1. Aufzeichnungsverfahren

Ein Magnetbandgerät hat näherungsweise eine Bandpaßcharakteristik. Gleichspannungen, die bei einer langen Folge von L-Bit oder H-Bit auftreten, lassen sich nicht aufzeichnen. Digitale Daten kann man mit einem Audiokassettengerät nur aufzeichnen, wenn sie codiert sind. Das Codierungsverfahren sollte dabei den Übertragungseigenschaften des Speichermediums gut angepaßt sein. Einige Möglichkeiten der Codierung werden nachfolgend kurz aufgezeigt.

Ein sehr einfaches Verfahren ist die Amplitudenmodulation. Die Datenbit tasten einen NF-Träger von etwa 2 kHz. Bild 8.1 zeigt das modulierte Signal. Die Information gewinnt man durch Gleichrichten und Sieben des wiedergegebenen Signals zurück. Eine praktische Realisierung wurde in [2] veröffentlicht. Nachteil dieses Verfahrens ist die nur geringe Datenübertragungsrate von (meist) nur 110 baud (110 Bit/s). Die Aufzeichnung eines Blocks von 1 kByte Länge dauert etwa 102 s.

Auch die Frequenzmodulation ist zum Codieren geeignet. Beispiele dafür sind das KIM-Interface und der Kansas-City-Standard. Bild 8.2 zeigt den Kansas-City-Standard. Ein L-Bit besteht

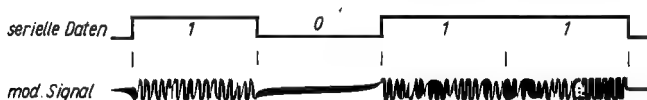


Bild 8.1 Aufzeichnung mittels Amplitudenmodulation

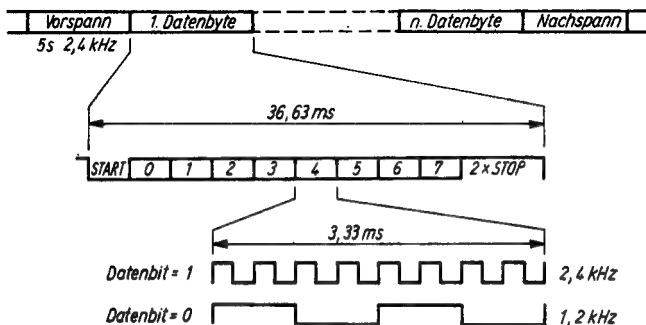


Bild 8.2 Aufzeichnung nach dem Kansas-City-Verfahren

aus 4 Schwingungen (1,2 kHz), ein H-Bit aus 8 Schwingungen (2,4 kHz). Moduliert wird z. B. mit einem FSK-Generator. Die Demodulation kann mit einem Impulsbreitendiskriminator durchgeführt werden. Ein Demodulator wurde in [5] beschrieben. Der Kansas-City-Standard ermöglicht eine recht sichere Datenaufzeichnung. Allerdings beträgt die Übertragungsgeschwindigkeit nur 300 baud, so daß ein Kansas-City-Interface für die Aufzeichnung großer Datenmengen ebenfalls nicht geeignet ist. Das Aufzeichnen eines 1-kByte-Blocks dauert etwa 38 s.

Weitaus höhere Datendichten erreicht man mit den verschiedenen PCM-Verfahren. Im Beispiel wurde Phase-Encoding (Richtungstaktschrift) verwendet. Bild 8.3 zeigt den Code sowie die Modula-

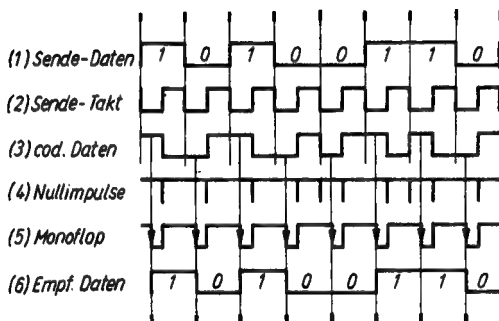


Bild 8.3 Aufzeichnung nach dem Phase-Encoding-Verfahren

tion und Demodulation. Eine Exklusiv-Oder-Verknüpfung von Takt (1) und Daten (2) bewirkt die Modulation. Die Impulsfolge (3) zeigt die codierten Daten: Ein H-Bit wird durch einen H/L-Sprung und ein L-Bit durch einen L/H-Sprung in der Mitte des Bit gekennzeichnet. Bei aufeinanderfolgenden gleichen Bit entstehen redundante Flußwechsel, die bei der Rückgewinnung der Information von den eigentlichen Flußwechseln unterschieden werden müssen. Das wird durch die Aufzeichnung von Synchronzeichen am Beginn eines Datenblocks erreicht. Bei der Wiedergabe gelangt das vom Kassettenrecorder kommende Signal an einen Nulldurchgangsdetektor. Die Nulldurchgänge (4) triggern einen monostabilen Multivibrator, der nach $3/4$ der Bitlänge wieder zurückkippt (5). Mit der entstehenden Flanke wird die Information in das Empfangsschieberregister (6) geschoben.

Das Kassetteninterface arbeitet mit einer Bitrate von 4800 baud. 1 kByte läßt sich in nur 2,25 s aufzeichnen. Das von den Autoren realisierte Interface wird mit dem Kassettenrecorder *Geracord GC-6020 portable* betrieben. Dieses Gerät verfügt über eine genau arbeitende Bandzähluhr, die das schnelle Auffinden von Datenaufzeichnungen ermöglicht. Auch andere Geräte wie *MIRA*, *MK 21* und *ZK 246* wurden an das Interface angeschlossen. Die Fehlerrate war bei Einsatz von hochwertigem Bandmaterial sehr gering. Treten bei der Wiedergabe dennoch Lesefehler auf, so sorgt die in Abschnitt 9. beschriebene Software für eine selbständige Fehlerkorrektur. Somit steht ein zuverlässiger, schneller und billiger Massenspeicher zur Verfügung.

8.2. Realisierung

Mit dem Interface lassen sich unterschiedliche Baudraten realisieren. Der etwas höhere Bauelementeaufwand bringt den zusätzlichen Vorteil, daß das Interface ohne komplizierten Abgleich in Betrieb genommen werden kann. Bild 8.4 zeigt die Schaltung des Kassetteninterface. Zentrale Bausteine sind die SIO *U 856* und der CTC *U 857* (siehe Abschnitt 7.1.). Das Kassetteninterface benutzt das Port A der SIO. Port A hat getrennte Eingänge für den Sendetakt (TxC) und den Empfangstakt (RxC), was für die Funktion der Schaltung nötig ist. Kanal 2 der CTC erzeugt den Takt. Gatter D1.3 arbeitet als Takttreiber für den Takt T.

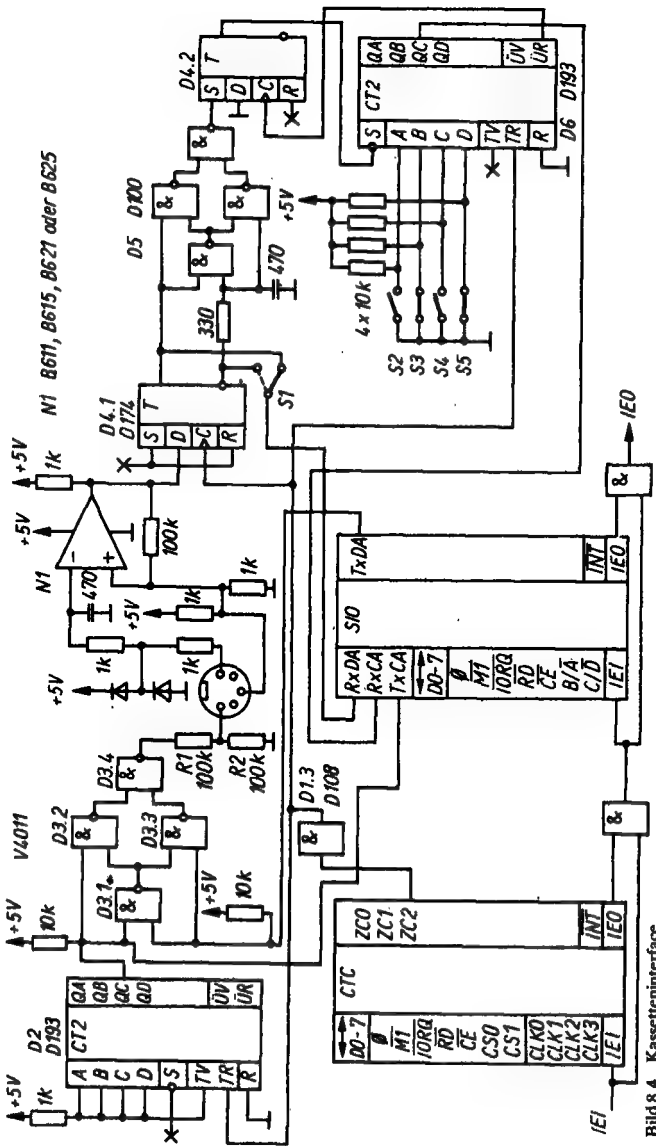


Bild 8.4 Kassetteninterface

Der Modulator besteht aus dem Zähler D2 (*D 193*) und einem Exklusiv-Oder, das aus den Gattern D3.1 . . . D3.4 gebildet wird. Der Zähler D2 teilt den Takt T durch 8. An QC des Zählers wird der Sendetakt TxCA abgenommen und der SIO zugeführt. Die Baudrate entspricht der Frequenz des Sendetaktes TxCA. Jede fallende Flanke des Sendetaktes schiebt 1 Bit aus dem Senderegister der SIO. Dieses Bit wird mit dem Takt TxCA exklusiv-oder-verknüpft. Am Ausgang von Gatter D3.4 steht das codierte Signal zur Verfügung. Über R1 und R2 gelangt es an den Kassettenrecorder. Der Demodulator ist etwas aufwendiger. Bild 8.5 zeigt das Impulsdiagramm.

Das Eingangssignal wird dem Operationsverstärker N1 zugeführt. An seinem Ausgang liegt das begrenzte Wiedergabesignal, welches durch das D-Flip-Flop D4.1 mit dem Takt T synchronisiert wird. Die synchronisierten Daten werden über den Schalter S1 an den Eingang RxDA der SIO gelegt. Der Schalter S1 legt die Phasenlage der Daten fest (je nach Kassettenrecorder). Bei falscher Phasenlage werden invertierte Daten gelesen. Die Stellung des Schalters muß man bei der Inbetriebnahme durch Versuch ermitteln. Den Empfangstakt RxCA erzeugen die Schaltkreise D4.2, D5 und D6. Der Zähler D6 und das D-Flip-Flop D4.2 bilden einen monostabilen Multivibrator. Die Gatter D5.1 . . . D5.4 erzeugen bei jedem Nulldurchgang des Eingangssignals einen kurzen L-Impuls. Dieser Impuls triggert den monostabilen Multivibrator. An Hand von Bild 8.6 soll die Funktionsweise erläutert werden. Am Ausgang Q des Flip-Flop D4.2 sei L-Pegel. Dieser Eingang ist mit dem Ladeeingang des Zählers D6 verbunden. Der an den Dateneingängen (mit den DIL-Schaltern S2 . . . S5) vorgewählte Wert wird in den Zähler übernommen. Der Nulldurchgangsimpuls setzt das Flip-Flop D4.2, Ausgang Q wird H. Mit dem nächsten Taktim-

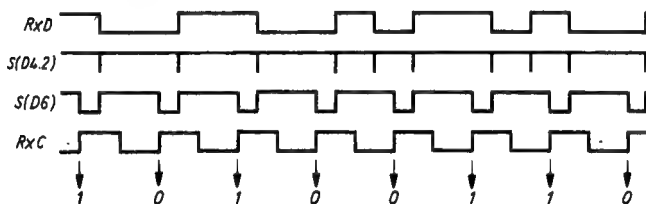
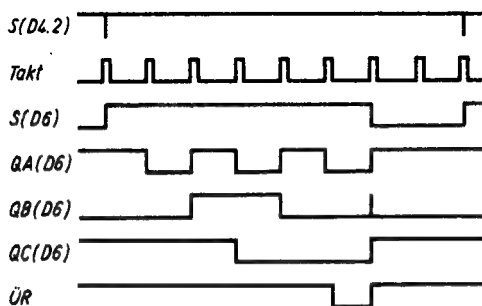


Bild 8.5 Impulsdiagramm des Demodulators



(nicht maßstäblich)

Bild 8.6 Impulsdiagramm des Monoflop

puls T beginnt der Zähler rückwärts zu zählen. Bei Erreichen des Zählerstandes „0“ entsteht ein Übertragsimpuls, der den Ausgang Q des Flip-Flop D4.2 auf L schaltet. Somit ist die Ausgangsstellung wieder erreicht. Am Ausgang Q des Flip-Flop entsteht die schon in Bild 8.3 dargestellte Impulsfolge (5). Den Empfangstakt RxCA für die SIO entnimmt man QC des Zählers D6. Mit jeder steigenden Flanke des Empfangstaktes übernimmt die SIO die an RxDA liegende Information in das Empfangsschieberegister. Die Lage dieser Flanke und damit die Impulsdauer des Monoflop werden mit den DIL-Schaltern S2 ... S5 auf 3/4 der Bitlänge eingestellt. Im Mustergerät lädt sich der Zähler mit dem Wert 5.

Die Inbetriebnahme ist mit der in Abschnitt 9. beschriebenen Software nicht schwierig. Es wird ein kurzer Datenblock aufgezeichnet. Werden die Daten bei Wiedergabe nicht gelesen, muß man die Stellung von Schalter S1 (Phasenlage) ändern. Der optimale Abtastzeitpunkt läßt sich dann mit S2 ... S5 einstellen (Richtwert: 5). Die Übertragungsgeschwindigkeit kann durch Verändern der Zeitkonstante des CTC-Kanals 2 den eigenen Wünschen angepaßt werden.

Tabelle 8.1 zeigt den Zusammenhang zwischen der zu programmierenden Zeitkonstante und der Baudrate bei Systemtaktfrequenzen von 2,5 und 2,4596 MHz. Die maximal mögliche Übertragungsrate hängt vom verwendeten Magnetbandgerät ab. Die Übertragungsgeschwindigkeit sollte nicht wesentlich kleiner als 1200 baud sein. Die im Mustergerät gewählte Baudrate von 4800 baud stellt einen guten Kompromiß zwischen möglichst geringer

**Tabelle 8.1. Zusammenhang: CTC-
Zeitkonstante-Baudrate**

TC	Baudrate	
	bei $f_c = 2,5 \text{ MHz}$	bei $f_c = 2,4576 \text{ MHz}$
1	19531	19200
2	9766	9600
3	6510	6400
4	4883	4800
5	3906	3840
6	3255	3200
8	2441	2400
12	1628	1600
16	1220	1200
32	610	600

Fehlerzahl und möglichst hoher Geschwindigkeit dar (1 kByte wird in 2,25 s aufgezeichnet). Hochwertiges Bandmaterial ist Grundvoraussetzung für die erreichten Werte.

Wie bereits erwähnt, werden eventuell auftretende Lesefehler selbständig korrigiert. Ermöglicht wird das durch eine geeignete Formatierung der Daten. Wie man die Daten formatiert, ist in Bild 8.7 dargestellt. Die aufzuzeichnenden Daten werden in einzelne Blöcke zu je 128 Byte (oder weniger, falls die Anzahl der zu übertragenden Byte < 128 beträgt) aufgeteilt. Ein Datenblock beginnt mit 6 Synchronzeichen. Es folgt das Zeichen GS (1 DH), das den Datenblockbeginn kennzeichnet. Das nächste Byte enthält die Anzahl der folgenden Datenbyte. Nach der Übertragung der n Datenbyte folgt das Prüfsummenbyte (die Übertragung von nur einem Prüfsummenbyte hat sich als ausreichend erwiesen).

Zu Beginn jeder Datei wird ein Block übertragen, der alle wichtigen Datenparameter enthält (Kopf). Der Kopf beginnt mit 20 Synchronzeichen. NL (01EH) kennzeichnet den Beginn der Aufzeichnung, wird also nur in Datei 1 (Bild 8.7) übertragen. Die Kennung des Kopfes ist das Zeichen SOH (01H). Die nächsten 8 Zeichen enthalten den Dateinamen. Die weiteren Parameter sind:

– TT Dateityp (maximal 2 Zeichen), zum Beispiel:

- 'P' Maschinenprogramm mit Autostart
- 'BA' BASIC-Programme
- 'A' Textdatei
- 'BI' Sonstige

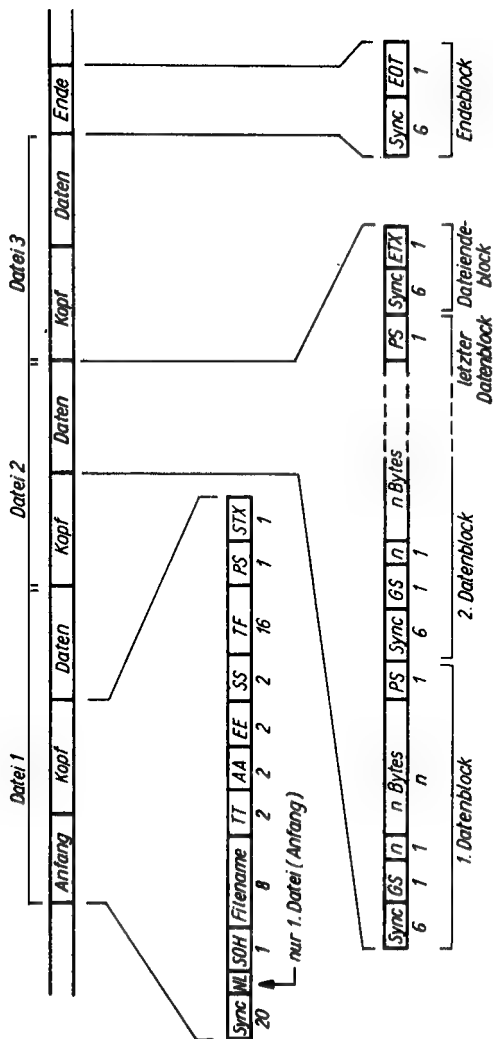


Bild 8.7 Verwendetes Dateiformat für Kassettenaufzeichnung

- AA Anfangsadresse (2 Byte)
- EE Endadresse (2 Byte)
- SS Startadresse für Autostart
- TF Textfeld (16 Zeichen)

Im Textfeld kann beispielsweise das Erstellungsdatum oder der Bearbeitungszustand eingetragen werden. Anschließend folgt die Prüfsumme des Kopfes. STX (02H) beendet den Kopfblock, zugleich zeigt STX den Beginn der Datenblöcke an. Sind alle Datenblöcke übertragen, erscheinen 6 Synchronzeichen und ETX (03H). Dieser kurze Block beendet die Datei. Die gesamte Datei wird danach noch 2mal übertragen. Die Aufzeichnung endet mit dem Endblock, der aus 6 Synchronzeichen und dem Zeichen EOT (04H) besteht.

Nicht nur die Hardware, auch die Software ist für die Wiedergabe umfangreicher als für den Aufzeichnungsvorgang. Eine Datei wird eingelesen, wenn der Dateiname der Aufzeichnung mit dem eingegebenen Dateinamen übereinstimmt. Dabei sind alle 8 Zeichen des Namens signifikant. Bei der Wiedergabe werden die einzelnen Datenblöcke mitgezählt. Traten keine Lesefehler auf, so ist nach Erreichen des Dateiendblocks (ETX) die Wiedergabe beendet. Lagen jedoch Lesefehler vor, so wird die nächste Datei gelesen und alle mit fehlerhafter Prüfsumme „gemerkten“ Datenblöcke werden neu geladen. Bei mangelhaftem Bandmaterial oder starken Gleichlaufschwankungen des Kassettenrecorders kann die Anzahl der Lesefehler so groß werden, daß eine Korrektur durch Neuladen der fehlerhaften Datenblöcke nicht mehr möglich ist. In diesem Fall lädt man die gesamte Datei neu. Da jede Datei 3mal aufgezeichnet wurde, ist in der Regel (auch bei weiteren Fehlern) das Laden (und Korrigieren) der Datei möglich. Wird beim Lesen der EOT-Block erreicht, so war keine fehlerfreie Wiedergabe möglich.

Die Software (Abschnitt 9.) organisiert den Datenverkehr. Details sind der Softwarebeschreibung und dem Listing zu entnehmen.

9. Das Monitorprogramm

Bisher wurde vor allen Dingen die Computerhardware beschrieben. Damit der Computer sinnvolle Funktionen ausführen kann, muß ihm ein geeignetes Programm eingegeben werden. Das in diesem Abschnitt beschriebene Monitorprogramm soll den Computer „zum Leben erwecken“.

Das Monitorprogramm benötigt als minimale Hardwarevoraussetzung die CPU-Karte (mit 3-kByte-ROM und 1-kByte-RAM), die Tastatur und die Bildschirmsteuerung. Wenn die I/O-Karte eingesetzt wird, sind alle vorhandenen Monitorkommandos ausführbar. Das Monitorprogramm wurde im wesentlichen unter folgenden Gesichtspunkten geschrieben:

- einfache Bedienung, Fehlertoleranz;
- einfache Ergänzung durch zusätzliche Programmmodule;
- einfache Arbeit mit externen Speichermedien;
- Schnittstellen zu höheren Programmiersprachen (Assembler, BASIC).

Der Monitor wird als ausführliches Assemblerlisting veröffentlicht. Dadurch lernt der Anwender zusätzlich die Funktionsweise kennen, er kann Änderungen einbauen, (zum Beispiel andere Adressen) oder einzelne Teile des Programms anderweitig verwenden.

9.1. Beschreibung der Monitorkommandos

Das Monitorprogramm erwartet nach der Ausgabe des Promptzeichens (&) die Eingabe einer Kommandozeile. Eine Kommandozeile besteht aus dem Kommando selbst und den dazugehörigen Angaben. Die einzelnen Angaben müssen durch ein Komma oder mindestens ein Leerzeichen getrennt werden. Die Eingabe wird durch RETURN (CR) abgeschlossen, der Computer führt nun das eingegebene Kommando aus.

Im folgenden sind die einzelnen Kommandos kurz beschrieben. Den Kommandonamen braucht man nicht auszuschreiben. So kann beispielsweise an Stelle von MEMORY auch MEMOR,

MEMO, MEM, ME oder M geschrieben werden. Alle angegebenen Zeichen sind signifikant. Beginnen verschiedene Kommandonamen mit dem gleichen Buchstaben und ist nur ein Buchstabe angegeben, so wird das zuerst gefundene Kommando angesprungen. M entspricht MEMORY, aber nicht MOVE, für MOVE muß mindestens MO eingegeben werden. Klarheit über die Reihenfolge der Kommandos schafft das Kommando HELP.

– **HELP** Anzeige aller im Speicher vorhandenen Kommandos
HELP CR

Sowohl die Kommandos des Monitors als auch die vom Anwender definierten werden gelistet. Im gesamten Speicher wird nach dem Kommandorahmen (0EDH, 0FFH, Kommandozeichenkette, 00H) gesucht, und die Kommandozeichenketten werden ausgegeben (siehe auch Abschnitt 9.2.).

– **MEMORY** Anzeigen und Modifizieren von Speicherbereichen/
MEMORY [Option] Startadresse CR

Es wird ein Speicherbereich ab Startadresse und 1/4 kByte Länge in hexadezimaler Form und in Textdarstellung ausgegeben. Bei der Textdarstellung erscheinen nicht darstellbare Codes als Punkt. Bei Angabe der Option @ A beginnt jede neue Zeile zusätzlich mit der entsprechenden Adresse. Dabei wird die Textdarstellung unterdrückt.

Mit dem MEMORY-Befehl lassen sich auch Speicherzellen modifizieren. Ein spezieller Cursor zeigt auf die zu modifizierende Speicherzelle. Die Cursorposition und damit die Adresse können mit den 4 Cursortasten (links, rechts, hoch, tief) verändert werden. Die Eingabe erfolgt hexadezimal (Mode H) oder als Textzeichen (Mode A). Zum Umschalten in den jeweils anderen Modus benutzt man die Escape-Taste (ESC).

Eine neue Startadresse läßt sich nach Drücken der Taste Q (Quit) eingeben. Folgt an Stelle einer Adresse ein weiteres Q, so wird das Programm MEMORY nach Drücken der RETURN-Taste verlassen.

– **MOVE** Umladen von Speicherbereichen

MOVE, Startadresse, Endadresse, Zieladresse CR

Es wird der Bereich von Startadresse bis Endadresse nach Zieladresse kopiert. Dabei können sich die Speicherbereiche auch überlappen. Nach Ausführung des MOVE-Befehls wartet das Programm auf eine Tastatureingabe: Q (Quit) oder RETURN (oder irgendeine andere Taste). Im letzteren Fall springt das Programm

nach MEMORY und zeigt den Speicherbereich ab Startadresse an.

- FILL Füllen eines Speicherbereichs mit einer Konstanten
 FILL, Startadresse, Endadresse [Byte] CR

Das angegebene Byte füllt den entsprechenden Speicherbereich. Fehlt das Füllbyte, so wird das Byte als 00H angenommen (schnelles Löschen eines Speicherbereichs).

- IN, OUT Bedienen von Ports
 IN, Portadresse CR
 OUT, Portadresse, Datenbyte CR

Mit dem IN-Befehl lassen sich Ports direkt lesen. Das gelesene Byte wird auf dem Consolkanal (Bildschirm) ausgegeben.

Mit dem OUT-Befehl können zum Beispiel Ports initialisiert werden.

- Beispiel: OUT 0 CR
 OUT 0 CR entspricht dem Kommando OFF

- OFF Ausschalten des Computers
 OFF CR

Nach etwa 20 s schaltet sich der Computer ab. Den Abschaltvorgang kann man durch RESET, NMI oder durch die Sequenz IN 0 CR, IN 0 CR abbrechen.

- SAVE Abspeichern auf Magnetband
 SAVE, Filename, Filetyp, Anfangsadr., Endadr.
 [Eintrittspunkt] [Textfeld] CR

SAVE zeichnet den Speicherbereich von Anfangs- bis Endadresse auf Magnetband auf. Der Name des Programms kann beliebig lang sein, es werden aber nur die ersten 8 Zeichen als Filename verwendet. Durch Angabe eines Filetyps läßt sich die Datei besonders kennzeichnen, beispielsweise können BASIC-Programme durch @ BA oder Textdateien durch @ A gekennzeichnet werden. Bei der Angabe eines Filetyps sind 2 Zeichen signifikant. Ist kein Filetyp angegeben, so wird vom SAVE-Programm automatisch P eingesetzt. P kennzeichnet die Datei als lauffähiges Maschinenprogramm. Dateien vom Typ P erfordern noch einen weiteren Parameter, nämlich den Eintrittspunkt des Programms. Bei nicht angegebenem Eintrittspunkt wird vom SAVE-Programm automatisch der Restart-Eintrittspunkt des Monitors eingetragen. Schließlich können im Textfeld noch zusätzliche Informationen, z. B. das Erstellungsdatum, folgen. Es werden 16 Textzeichen (nach;) aufgezeichnet. Dieses Textfeld werten die hier beschriebenen Monitor-kommandos nicht aus. Der Anwender kann den Monitor aber

durch weitere Kommandos ergänzen, die bei Bedarf dieses Textfeld auch auswerten (von den Autoren wurde ein Programm CAT geschrieben, das alle auf Band befindlichen Dateien mit ihren Eigenschaften einschließlich Textfeld auflistet).

- **LOAD** Laden von Magnetband
 LOAD, Filename [Option] [Offset] CR

Ein mit **SAVE** aufgezeichneter Speicherbereich wird wieder eingelesen. Wenn der angegebene Filename mit dem aufgezeichneten Namen übereinstimmt, wird die Datei geladen. Man erkennt das an der Aufschrift „**FILE FOUND**“. Nach dem Laden, falls es sich um ein abarbeitbares Maschinenprogramm (Filetyp **P**) handelt, startet das Programm automatisch (Eintrittsadresse), es sei denn, es wurde die Option **@ Q** angegeben. **@ Q** unterdrückt den Autostart. Bei Bedarf kann noch ein Offsetwert angegeben werden. Der Wert addiert sich zur Ladeadresse und ermöglicht das Laden auf beliebige Speicherbereiche. Dabei ist natürlich der eventuelle Autostart unterdrückt. Nach dem Laden werden die eventuell aufgetretenen Lesefehler angezeigt. Diese Fehler wurden aber vom **LOAD**-Programm korrigiert. Nur wenn „**BAD FILE**“ erscheint, war keine Korrektur möglich, d. h., die im **RAM** befindlichen Daten sind fehlerhaft.

- **READ** Lochstreifen lesen
 READ [Option] [Offset] CR

Dieser Befehl ermöglicht das Lesen von **MC-Lochstreifen** (**Intel-Hex-Format**) vom Lochstreifenlesekanal **RI**. Wird die Option **@ A** angegeben, dann stoppt das Programm nach Einlesen 1. Adresse, gibt diese Adresse aus und wartet auf eine Eingabe. Drückt man ohne Eingabe **RETURN**, so wird, beginnend bei der angezeigten Startadresse, eingelesen. Es kann aber auch eine andere Adresse eingegeben werden. Nach dem Drücken der **RETURN**-Taste wird dann das Einlesen, beginnend bei der neuen Adresse, fortgesetzt. Eine andere Möglichkeit, den Lochstreifen „verschoben“ einzulesen, besteht darin, daß man einen Offsetwert angibt.

Beim Auftreten eines Lesefehlers stoppt das Programm. Mit Hilfe der Taste **Q** kann nun das **READ**-Programm verlassen werden. Es ist aber auch möglich, den fehlerhaften Block noch einmal einzulesen (Lochstreifen zum Blockanfang [:] zurücktransportieren, dann **RETURN**).

- NMI Vektor eintragen
NMI CR

Es wird auf die Adresse 066H ein Rücksprung zum Monitor eingetragen. Den nichtmaskierten Interrupt kann man dann als „Notbremse“ nutzen. Die CPU-Register werden gerettet. Ein ununterbrochenes Programm wird durch GO PC mit den geretteten Registerwerten wieder gestartet. Dabei ist zu beachten, daß sich nur Programme wieder starten lassen, die nicht den Systemstack verwenden.

- GO Start von Programmen
GO, Startadresse [Breakpoint] CR
GO PC CR

Mit GO adr. läßt sich ein Programm starten. Der Stackpointer wird auf die Adresse USTCK (Userstack) gesetzt. Wenn das Anwenderprogramm diesen Stack benutzt, kann es durch RET (0C9H) in den Monitor zurückkehren. Es ist möglich, zusätzlich den Breakpoint zu setzen, wenn das Anwenderprogramm zu Testzwecken unterbrochen werden soll. Der Breakpoint muß auf den Anfang eines Befehls gesetzt werden. Das GO-Programm rettet das dort stehende Byte und trägt 0FFH (Restart 38 H) ein. Bei Erreichen des Breakpoints wird das ursprüngliche Byte wieder zurückgeladen, anschließend werden alle CPU-Register angezeigt. Sie lassen sich nun gegebenenfalls verändern (REG). Das Anwenderprogramm wird durch GO PC fortgesetzt. Bei der Anwendung von GO PC wird der RAM-Bereich der Userregister in die CPU kopiert (siehe REG). Jedes Programm kann statt mit GO adr. auch mit GO PC gestartet werden (SP berücksichtigen!). GO ohne weitere Angaben führt keinen Programmstart aus, sondern eignet sich dazu, einen nicht erreichten Breakpoint zu löschen.

- REG Anzeigen und Modifizieren der CPU-Register
REG [reg Byte [reg Byte . . .]] CR

Das Kommando REG ohne weitere Angaben zeigt die CPU-Register an. Die Register verändert man durch die Eingabe des Registers und des gewünschten Wertes. Beispielweise bewirkt das Kommando

REG A0, SP1000, PC100, H'0

die Belegung der Register A, H', SP und PC mit den angegebenen Werten. Genaugenommen werden die entsprechenden RAM-Zellen (Userregister UREG) verändert.

– **CHECKSUM** Prüfsummenberechnung

CHECKSUM, Startadr., Endadr. CR

Dieses Kommando berechnet von dem angegebenen Speicherbereich eine 2-Byte-Prüfsumme.

– = Adreßrechnung

= , x, y, CR

Diese Eingabe bewirkt folgende Berechnung und Ausgabe:

SUM : x + y DIF : x – y REL: **

Dabei bedeutet REL die relative Distanz bei Sprungbefehlen (JR, DJNZ usw.). Ist die Distanz zu groß, so werden an Stelle eines Byte 2 Sterne ausgegeben.

– **FIND** Suchen von Bytes oder Zeichenketten

FIND, Startadr., Endadr., Byte 1, Byte 2 . . . Byte n

CR FIND, A, Startadr., Endadr., Zeichenkette CR

FIND ermöglicht das Suchen nach einer Bytefolge oder nach einer Zeichenkette in dem Speicherbereich von Start- bis Endadresse. Auf dem Bildschirm werden alle Adressen ausgegeben, auf denen die Zeichenkette oder Bytefolge gefunden wurden.

– **SIZE** Speichertest

SIZE CR

SIZE führt einen einfachen, nicht zerstörenden RAM-Test durch, gibt den höchsten als RAM verfügbaren Speicherplatz aus und erwartet eine Eingabe. Die ermittelte Adresse oder eine jetzt einzugebende Adresse wird nach RETURN in die Zelle MSIZE geladen. Der Monitor selbst benötigt die Angabe über den höchsten Speicherplatz nicht. Aber verschiedene Programme (z. B. BASIC-Interpreter) brauchen in der Initialisierungsphase die Angabe über den höchsten zur Verfügung stehenden Speicherplatz, der entweder das RAM-Ende oder eine andere Adresse sein kann.

– **EOF** Dateiendezeichen definieren

EOF CR

Verschiedene Programme (BASIC-Interpreter, Assembler usw.) benötigen beim Einlesen von Daten und Programmen ein End-of-File-Zeichen. Fehlt dieses Zeichen auf dem Datenträger (z. B. Lochstreifen), so läßt sich die Lesefunktion dieser Programme nicht beenden. Um dieses Problem zu lösen, wurde der Monitor so ausgelegt, daß nach Drücken von CTRL C EOF-Zeichen gesendet werden. Da die verschiedenen Programme unterschiedliche EOF-Zeichen voraussetzen, wurde dieses Kommando geschaffen. Es

zeigt das derzeit gültige EOF-Zeichen an. Das Zeichen kann wieder übernommen oder durch Neueingabe verändert werden.

- ASN Gerätezuweisung
 ASNRI = RI CR
 ASNRI = CRI CR
 ASNRI = USR, Adresse CR

Mit diesem Kommando kann man die jeweilige Gerätezuweisung verändern (siehe auch Abschnitt 9.2.). Hat beispielsweise ein Anwenderprogramm nur eine Lochstreifenschnittstelle, so lassen sich die Daten bei der Zuweisung RI = RI vom Lochstreifenleser oder bei RI = CRI vom Kassettenmagnetbandgerät einlesen.

Im vorliegenden Monitor sind nur wenige Zuweisungen fest vorhanden. Eine Erweiterung ist leicht möglich (siehe Abschnitt 9.2.). Man muß allerdings beim Verändern der Consolzuweisung aufpassen, sonst wird das Bedienen des Computers in Frage gestellt (Die Gerätezuweisungen wirken auch auf den Monitor!). In solchen Fällen hilft dann nur noch RESET.

9.2. Schnittstellen und Erweiterbarkeit

Der Monitor hat ab Adresse 0F000H einige Eintrittspunkte, die im folgenden kurz erläutert werden:

JMP BEGIN Neustart des Monitors

JMPCI Consoleingabe (Tastatur, Zeichen in A)

JMPRI Lesekanal (Lochstreifenleser, Zeichen in A)

JMPCOE Consolausgabe (Bildschirm, Zeichen C)

JMPPOE Punchkanal (Lochstreifenstanzer, Zeichen in C)

JMPLOE Listkanal (Drucker, Zeichen in C)

JMPCSTS Consolstatus (A = 0 keine Taste, A = 0FFH gedrückt)

JMPCRI Kassettenlesekanal (Zeichen in A)

JMPCPOE Kassettenpunchkanal (Zeichen in C)

JMPMEMSI höchster Speicherplatz (A = Low, B = High-Teil der Adresse)

JMPMAIN Restart des Monitors

JMP EXT Aufruf von Monitorunterprogrammen (C = Nr. des Programms entsprechend ETAB)

DA FINA Beginn RAM-Bereich des Monitors.

Ein Teil dieser Sprünge führt in den RAM-Bereich des Monitors.

Von dort wird dann zur eigentlichen Geräteroutine gesprungen. Dadurch ist eine einfache Gerätezuweisung möglich. Beim Neustart benutzt man folgende Zuordnung (ATAB):

Consoleingabe	CI	— CI	Tastatur
Lesekanal	RI	— RI	Lochstreifenleser
Consolausgabe	COE	— CO	Bildschirm
Punchkanal	POE	— CPO	Kassettenausgabe
Listkanal	LOE	— CO	Bildschirm
Kassettenlesekanal	CRI	— CRI	Kassetteingabe
Kassettenpunchkanal	CPO	— CPO	Kassettenausgabe

Bei Veränderung durch ASN wird auf den entsprechenden RAM-Zellen nur die jeweilige Gerätetreiberadresse eingetragen. Weitere Einsprungstellen (wenn Zeichen im Akkumulator übergeben werden sollen) sind:

FINA + 0B6H Listkanal

FINA + 0BAH Punchkanal

FINA + 0BEH Kassettenpunchkanal

FINA + 0C2H Consolausgabekanal

Wie man die Schnittstelle benutzt und den Monitor erweitert, belegen 3 einfache Beispiele.

Im 1. Fall sollte eine Schreibmaschinenfunktion realisiert werden. Der Aufrufer erfolgt durch Eingabe Typ und RETURN

	DB 0EDH	; Kennzeichnungsrahmen
	DB 0FFH	; wird von MAIN gesucht
	DB 'TYP'	; Kommandoname
	DB 00H	; Ende des Rahmens
LOOP:	CALL 0F003H	; Zeichen von Tastatur holen
	CMP 03H	; CTRL C gedrückt?
	Rz	; Ja, zum Monitor zurück
	LD C, A	
	CALL 0F009H	; Zeichen ausgeben
	JR LOOP - #	

Der Monitor (Routine MAIN) sucht den Kennzeichnungsrahmen. Wenn die Bytefolge 0EDH und 0FFH gefunden wurde, wird die folgende Zeichenkette mit der im Textpuffer abgelegten Kommandozeichenkette verglichen. Sind beide Zeichenketten identisch und folgt im Speicher das Byte 00H, so wird die dem Rahmenendebite folgende Adresse von MAIN aus angesprungen. In diesem Beispiel handelt es sich um die Marke LOOP. Dabei ist es gleichgültig, wo die Folge 0EDH, 0FFH, Zeichenkette, 00H im Speicher

steht. Der gesamte Speicher wird nach dem eingegebenen Kommando durchsucht. Die Routine MAIN beginnt die Suche ab Marke START – 1000H (0E000H). Das bedeutet, daß beispielsweise eine veränderte MEMORY-Routine ab Adresse 0E000H abgelegt und bei Eingabe des Befehls MEMORY aufgerufen werden könnte. Die im Monitor-EPROM enthaltene Routine MEMORY wird in diesem Fall nicht mehr gefunden. Jedes Anwenderprogramm kann man mit dem (im Listing HEAD genannten) Rahmen, dem Namen und anschließenden Byte 00H versehen. Auf diese einfache Art und Weise ist es möglich, jedes Programm mit seinem Namen vom Monitor aus aufzurufen. Die Rückkehr zum Monitor erfolgt durch einen RET-Befehl.

Im 2. Beispielsollen alle Zeichen, die über den Listkanal an den Monitor gelangen, auf Magnetband aufgezeichnet werden. Benötigt wird dazu eine neue Gerätezuweisung LO=CPO.

DB 0EDH

DB 0FFH

DB 'ASNLO=CPO'

DB 00H

LD HL, CPO1 ; Adresse Kassettentreiber

LD (LO+1), HL ; ins RAM laden

RET ; zurück zum Monitor

Im 3. Beispiel soll der Hexadezimalcode eines von der Tastatur eingegebenen Zeichens auf dem Bildschirm ausgegeben werden.

ENTRY: CALL 0F003H ; Zeichen von Tastatur holen

PUSH AF ; retten

LD C, A

CALL 0F009H ; Zeichen ausgeben

LD C, 5 ; Code für PRINT

CALL 0F021H ; EXT anspringen

DB ' = ' ; Zeichenkette ausgeben

DB 00H ; bis 0

POP AF ; Zeichen zurück

LD C, 3 ; Code für HEX-Ausgabe

CALL 0F021H ; EXT anspringen

RET ; zurück zum Monitor

Dieses kurze Programm zeigt, wie man den Programmteil EXT anwendet. Die Codes enthält Tabelle ETAB (siehe Listing). Auch die Funktionsweise der einzelnen Unterprogramme ist dem Listing zu entnehmen.

Die 3 Beispielprogramme sollten die Anwendung der Monitorroutinen demonstrieren. Weitere Einzelheiten findet man im ausführlichen Assemblerlisting

9.3. Das Programm

```

EBAS-88: R

PASS: P      EBAS-88      26.1.84

LOC      CODE      LINE      SOURCE

00001      PN      MON
00002
00003      *****
00004      * MONITORPROGRAMM
00005      *
00006      * V2.1 20.12.83
00007      * ROM-TEIL
00008      *****
00009
00010
00011      ORG      OF000H
00012
00013      ;SPRUNGTABELLE:
00014
00015      START:  JMP      BEGIN
00016      F003    C3CBFF  JMP      CI
00017      F006    C3C5FF  JMP      RI
00018      F009    C3C1FF  JMP      COE
00019      F00C    C3B9FF  JMP      POE

F000    C326F0
F003    C3CBFF
F006    C3C5FF
F009    C3C1FF
F00C    C3B9FF

;TASTATUR
;LOCHBANDLESER
;BILDSCHIRM
;LOCHBANDSTANZER

```


F059	CD74F1	00050	ERROR: CALL PRINT	
F05C		00051	DE 'ILLEGAL COMMAND'	
F06B	00	00052	DE 0	
		00053		
		00054		
		00055	; BEFEHLSZEILE EINGEBEN UND PROGRAMM ANSPRINGEN	
		00056		
F06C	31FEPE	00057	MAIN: LD SP, STACK	; NEU LADEN
F06F	CD81F1	00058	CALL INL	; EINGABE BEFEHLSZEILE
F072	CD55F1	00059	CALL FNEXT	; SPACE AUSBLENDEN
F075	38F5	00060	JRC MAIN-#	; NUR BEI CR
F077	ED5340FF	00061	LD (TXTPT), DE	
F07B	2100E0	00062	LD HL, START-1000H	; SUCHBEGINN
F07E	01FFFF	00063	LD BC, OFFFPH	; 64K DURCHSUCHEN
F081	3EED	00064	MAIN4: LD A, L(HEAD)	
F083	EDB1	00065	CPH	; NACH ED SUCHEN
F085	20D2	00066	JRNZ ERROR-#	; FEHLER
F087	3EPF	00067	LD A, H(HEAD)	; FOLGT FF?
F089	BE	00068	CMP (HL)	; NEIN
F08A	20F5	00069	JRNZ MAIN1-#	
F08C	ED5B40FF	00070	LD DB, (TXTPT)	
F090	1A	00071	LD A, (DE)	; VERGLEICHEN
F091	23	00072	INC HL	
F092	13	00073	INC DB	
F093	0E	00074	DEC BC	
F094	BE	00075	CMP (HL)	
F095	28F9	00076	JRZ MAIN2-#	; GLEICH
F097	FE20	00077	CMP ', '	; TEST TRENNZEICHEN/CR
F099	2808	00078	JRZ MAIN3-#	
F09B	FE2C	00079	CMP ', '	

FO9D 2804	JRZ MAIN3-#	00080	
FO9F FE0D	CMP CR	00081	
FOA1 20DE	JRNZ MAIN1-#	00082	
	;EVT.DOCH KEIN BEFEHL?	00083	
	MAIN3: XOR A	00084	
FOA3 AF	DEC HL	00085	
FOA4 2B	PUSH BC	00086	
FOA5 C5	LD BC,8	00087	
FOA6 010800	CPIR	00088	
FOA9 EDB1	POP BC	00089	
FOAB C1	JRNZ MAIN1-#	00090	
FOAC 20D3	HL ZEIGT JETZT AUF 1.BEFEHL	00091	
	;DE ZEIGT JETZT AUF 1.ARGUMENT	00092	
	LD BC,MAIN	00093	
FOAE 016CF0	PUSH BC	00094	
FOB1 C5	JMP (HL)	00095	
FOB2 E9		00096	
		00097	
	;EINZELNE ARGUMENTE HOLEN	00098	
		00099	
FOB3 CD55F1	GFINA: CALL FNEXT	00100	
FOB6 2100FF	LD HL,FINA	00101	
FOB9 0608	LD B,8	00102	
FOBB CD55F1	GFIN1: CALL FNEXT	00103	
FOBE 280C	JRZ GFIN2-#	00104	
F0C0 77	LD (HL),A	00105	
F0C1 23	INC HL	00106	
F0C2 13	INC DE	00107	
F0C3 10F6	DJNZ GFIN1-#	00108	
FOC5 CD55F1	GFIN3: CALL FNEXT	00109	
	;WEITERSUCHEN		
	;TEST RAHMEN		
	;RUECKSPRUNG		
	;SPRUNG ZUM PROGRAMM		
	;FILENAME		
	;WEIL TRENNZEICHEN		
	;UEBERFLUESSIGE ZEICHEN IGNORIEREN		

FOC8	00110	RZ	DE		
FOC9 13	00111	INC	GPIN3-#		
FOCA 18F9	00112	JR	(HL),'		
FOCC 3620	00113	LD	HL		
FOCE 23	00114	INC	GPIN2-#		
FOCF 10FB	00115	DJNZ			
FOD1 C9	00116	NIGHT: RET			
	00117				
FOD2 CD55P1	00118	GPITY: CALL FNEXT			
FOD5 2108FF	00119	LD HL,PITY			
FOD8 0602	00120	LD B,2			
FODA FE40	00121	CMP 'Q'			
FODC 20EE	00122	JRNZ GPIN2-#			
FODE 13	00123	INC DE			
FODF 18DA	00124	JR GPIN1-#			
	00125				
FOE1 CD55P1	00126	GCOW: CALL FNEXT			
FOE4 2110FF	00127	LD HL,DATA			
FOE7 0610	00128	LD B,16			
FOE9 FE3B	00129	CMP ';			
FOEB 20DF	00130	JRNZ GPIN2-#			
FOED 13	00131	GCOW1: INC DE			
FOEE 1A	00132	LD A,(DE)			
FOEF FE0D	00133	CMP CR			
FOF1 28D9	00134	JRZ GPIN2-#			
FOF3 77	00135	LD (HL),A			
FOF4 23	00136	INC HL			
FOF5 10F6	00137	DJNZ GCOW1-#			
FOF7 C9	00138	RET			
	00139				

; MIT SPACE AUFPUELLEN

; FILETYP/OPTION

; ZUSAEZL. ANGABEN

POP8	3E20	00140	;ASCII -> HEX		
POP9	CDC2FF	00141		LD A, ' '	;HEX VON TASTATUR HOLEN
POP10	CD87F1	00142	INHEX:	CALL CO	
		00143		CALL INLOO	
		00144			
		00145			
P100	CD55F1	00146	ASHEX:	CALL FNEXT	;KONVERTIEREN
P103	CD20F1	00147		CALL CNVBN	
P106	1A	00148		LD A, (DE)	
P107	D8	00149		RC	;KEIN HEX
P108	C5	00150		PUSH BC	
P109	210000	00151		LD HL, 0	
P10C	1A	00152	ASH1:	LD A, {DE}	
P10D	CD20F1	00153		CALL CNVBN	
P110	380B	00154		JRC ASH4-#	
P112	29	00155	ASH3:	ADD HL, HL	;SCHIEBEN
P113	29	00156		ADD HL, HL	
P114	29	00157		ADD HL, HL	
P115	29	00158		ADD HL, HL	
P116	0600	00159		LD B, 0	
P118	4F	00160		LD C, A	
P119	09	00161		ADD HL, BC	
P11A	13	00162		INC DE	
P11B	18EF	00163		JR ASH1-#	
P11D	C1	00164	ASH4:	POP BC	
P11E	AF	00165		XOR A	
P11F	C9	00166		RET	
		00167			
P120	D630	00168	CNVBN:	SUB 30H	
P122	D8	00169		RC	

F123	FE0A	00170	CMP	10
F125	3F	00171	CCF	
F126	D0	00172	RNC	
F127	FE11	00173	CMP	11H
F129	D8	00174	RC	
F12A	FE17	00175	CMP	17H
F12C	3F	00176	CCF	
F12D	D8	00177	RC	
F12E	D607	00178	SUB	7
F130	C9	00179	RET	
		00180		
		00181		
		00182		
		00183	;OUTHEX	
		00184	;WANDELT HEX->ASCII	
		00185	;OUTH GIBT BYTE <A> AUS	
		00186		
F131	F5	00187	OUTH:	PUSH AF
F132	1F	00188		
F133	1F	00189		
F134	1F	00190		
F135	1F	00191		
F136	CD3AF1	00192	CALL	OUTH1
F139	F1	00193	POP	AF
F13A	F5	00194	OUTH1:	PUSH AF
F13B	E60F	00195	AND	00001111B
F13D	C630	00196	ADD	030H
F13F	FE3A	00197	CMP	03AH
F141	3802	00198	JRC	OUTH2-#
F143	C607	00199	ADD	07
F145	CDC2FF		OUTH2:	CALL C0

;SCHIEBEN

F148	F1	POP	AF	
F149	C9	RET		
				; OUTHL GIBT WORT <HL> AUS
F14A	F5	OUTH:	PUSH AF	
F14B	7C	LD	A, H	
F14C	CD31F1	CALL	OUTH	
F14F	7D	LD	A, L	
F150	CD31F1	CALL	OUTH	
F153	F1	POP	AF	
F154	C9	RET		
F155	1A	FNEXT:	LD A, (DE)	
F156	FE20	CMP	' ,	
F158	200D	JRNZ	FNEX3-#	
F15A	13	FNEX1:	INC DE	
F15B	1A	LD	A, (DE)	
F15C	FE20	CMP	' ,	
F15E	28FA	JRZ	FNEX1-#	
F160	FE0D	CMP	CR	
F162	2805	JRZ	FNEX4-#	
F164	AF	FNEX2:	XOR A	
F165	1A	LD	A, (DE)	
F166	C9	RET		
F167	FE0D	FNEX3:	CMP CR	
F169	37	FNEX4:	SCF	
F16A	C8	RZ		
F16B	3F	CCF		
F16C	FE2C	CMP	' ,	

; SPACE AUSBLENDEN

[illegible]

[illegible]

P272	CEBC	00380	RES	7, H		
P274	7C	00381	LD	A, H		
P275	FE51	00382	CMP	'Q'		;MODE
P277	2006	00383	JRNZ	MEM73-#		
P279	CDF8F0	00384	CALL	INHEX		
P27C	C3CFF1	00385	JMP	MEM01		;NEUE ADRESSE
P27F	D5	00386	PUSH	DE		
P280	5C	00387	LD	E, H		;ENDADR.+1 MERKEN
P281	1602	00388	LD	D, 2		;MODE FUER SPAETER
P283	CDCBFF	00389	CALL	CI		;TETRADEN BYTEINGABE
P286	FE1B	00390	CMP	ESC		;ZEICHEN VON TASTATUR HOLEN
P288	2015	00391	JRNZ	MEM10-#		;UMSCHALTUNG HEX<==>ASCII EINGABE
P28A	3E48	00392	LD	A, 'H'		
P28C	BE	00393	CMP	E		;VERGLEICH MIT MODE
P28D	2002	00394	JRNZ	MEM9-#		;H ==>MODE
P28F	3E41	00395	LD	A, 'A'		
P291	5F	00396	LD	E, A		;;
P292	3A10FF	00397	LD	A, (MEMMO)		;;
P295	E680	00398	AND	80H		;;
P297	B3	00399	OR	E		;;
P298	3210FF	00400	LD	(MEMMO), A		;KORR.
P29B	D1	00401	POP	DE		
P29C	C3F0F1	00402	JMP	MEM1		
P29F	01F0FF	00403	LD	BC, -MEMB		;INC FUER CURSOR
P2A2	FE0B	00404	CMP	CURU		;HOCH
P2A4	2836	00405	JRZ	MEM15-#		
P2A6	0EFF	00406	LD	C, -1		;BC=-1
P2A8	FE08	00407	CMP	CURL		;LINKS
P2AA	2830	00408	JRZ	MEM15-#		
P2AC	03	00409	INC	BC		

F2AD	03	00410	INC BC	BC=1
F2AE	FE09	00411	CMP CURR	;RECHTS
F2B0	282A	00412	JRZ MEM15-#	
F2B2	0E10	00413	LD C, MEMB	
F2B4	FE0A	00414	CMP CURD	;RUNTER
F2B6	2824	00415	JRZ MEM15-#	
F2B8	FE51	00416	CMP 'Q'	;QUIT?
F2BA	28D5	00417	JRZ MEM9-#	
		00418	;RAM BESCHREIBEN	
F2BC	6F	00419	MEM11: LD L, A	;ZEICHEN RETTEN
F2BD	7B	00420	LD A, E	;MODE
F2BE	FE41	00421	CMP 'A'	;ASCII EINGABE?
F2C0	7D	00422	LD A, L	
F2C1	2810	00423	JRZ MEM14-#	;JA
		00424	; HEX EINGABE	
F2C3	CD20F1	00425	CALL CNVBN	;KEIN HEX, NEUE EINGABE
F2C6	38BB	00426	JRC MEM8-#	;HILFSZELLEN
F2C8	210EFF	00427	LD HL, ARG3	
F2CB	ED6F	00428	HLD	
F2CD	15	00429	DEC D	;TETRADENZÄHLER
F2CE	20B3	00430	JRNZ MEM8-#	;NÄCHSTES ZEICHEN
F2D0	3A0EFF	00431	MEM13: LD A, (ARG3)	;NEUES BYTE
F2D3	2A0AFF	00432	MEM14: LD HL, (ARG1)	;CURSOR
F2D6	77	00433	LD HL, A	;=>RAM
F2D7	BE	00434	CMP (HL)	;ANGEKOMMEN?
F2D8	202B	00435	JRNZ MEM19-#	;NEIN, RAM FEHLER
F2DA	0E01	00436	LD C, 1	;BC=1;ADR INC
F2DC	D1	00437	MEM15: POP DE	;ENDE+1
F2DD	2A0AFF	00438	LD HL, (ARG1)	;CURSOR
F2E0	09	00439	ADD HL, BC	;CURSOR+INC

F31A	00	00470	DB	'MOVE'
F31E	00	00471	NOP	
F31F	CD00F1	00472	CALL	ASHEX
F322	DA59F0	00473	JPC	ERROR
F325	220AFF	00474	LD	(ARG1),HL
F328	CD00F1	00475	CALL	ASHEX
F32B	DA59F0	00476	JPC	ERROR
F32E	220CFF	00477	LD	(ARG2),HL
F331	CD00F1	00478	CALL	ASHEX
F334	DA59F0	00479	JPC	ERROR
F337	44	00480	LD	B,H
F338	4D	00481	LD	C,L
F339	2A0AFF	00482	LD	HL,(ARG1)
F33C	ED5B0CFF	00483	LD	DE,(ARG2)
F340	E5	00484	PUSH	HL
F341	AF	00485	XOR	A
F342	EB	00486	EX	DE,HL
F343	ED52	00487	SBC	HL,DE
F345	23	00488	INC	HL
F346	50	00489	LD	D,B
F347	59	00490	LD	E,C
F348	44	00491	LD	B,H
F349	4D	00492	LD	C,L
F34A	E1	00493	POP	HL
F34B	E5	00494	PUSH	HL
F34C	E7	00495	OR	A
F34D	ED52	00496	SBC	HL,DE
F34F	E1	00497	POP	HL
F350	3804	00498	JRC	MOVE1-#
F352	EDB0	00499	LDIR	

MOVE:

; ENDE-ANF=ANZAHL
; DE-ZIEL
; BC-LAENGE
; HL-QUELLE
; CY=0
; MOVE-RICHTUNG
; VORWAERTS UMLADEN

```

F354      1808
F356      0B
F357      09
F358      EB
F359      09
F35A      EB
F35B      03
F35C      EDB8
F35E      CDBFF
F361      2A0AFF
F364      C3CFF4

00500      JR      MOVE2-#
00501      DEC     BC
00502      ADD     HL,BC
00503      EX      DE,HL
00504      ADD     HL,BC
00505      EX      DE,HL
00506      INC     BC
00507      LDDR
00508      MOVE2:  CALL CI
00509      LD      HL,(ARG1)
00510      JMP     MEM01
00511
00512
00513
00514      ;*  LOAD
00515      ;*****
00516
00517      ;D7=1.LESEN,D6=UEBERLESEN
00518      ;D5=BLOCK WAR FALSCH,D4=ALLES KORRIGIERT
00519
00520      DA      HEAD
00521      DB      'LOAD'
00522      NOP
00523      LDEXT:  CALL GFINA
00524      CALL     GFITY
00525      CALL     ASHEX
00526      JRNC    LOAD-#
00527      LD      HL,0
00528      LOAD:   PUSH HL
00529      POP      IY

```

; QUIT/MEM

; KEIN OFFSET

F37F	CD61FA	00530	CALL	CINIT		
F382	2A08FF	00531	LD	HL, (FITY)		
F385	1E00	00532	LD	E, 0		: OPTION
F387	D9	00533	EXX			; B-FEHLER, D=KORREKTURZAHLER
F388	1680	00534	L0D0:	D, 80H		: 1. LESEN
F38A	CB7A	00535	L0D1:	BIT 7, D		
F38C	CA8CF4	00536	JPZ	L0D12		; FILE NICHT LESBAR
F38F	D9	00537	EXX			
F390	1600	00538	LD	D, 0		
F392	D9	00539	EXX			
F393	CDADP4	00540	CALL	L0D14		; SYNCHRONISIEREN
F396	FE1E	00541	CMP	NL		
F398	2003	00542	JRNZ	L0D3-#		
F39A	CDC8FF	00543	CALL	CRI		
F39D	FE01	00544	CMP	SOH		
F39F	20P2	00545	JRNZ	L0D2-#		; KEIN SOH
F3A1	AF	00546	XOR	A		
F3A2	4F	00547	LD	C, A		; PRUEFSUMME=0
F3A3	5F	00548	LD	E, A		; BLOCKZAHLER
F3A4	2100FF	00549	LD	HL, FINA		
F3A7	0608	00550	LD	B, 8		
F3A9	CDC8FF	00551	CALL	CRI		; FILENAME VERGLEICHEN
F3AC	BE	00552	CMP	(HL)		
F3AD	20DB	00553	JRNZ	L0D1-#		; NICHT GLEICH
F3AF	23	00554	INC	HL		
F3B0	81	00555	ADD	C		
F3B1	4F	00556	LD	C, A		
F3B2	10F5	00557	DJNZ	L0D4-#		; KOPF LADEN
F3B4	0618	00558	LD	B, 24		; 1. LESEN
F3B6	CB7A	00559	BIT	7, D		

F3B8	2002	00560	JRNZ LOD5-#		
F3BA	CBF2	00561	SET 6, D		
F3BC	CDBFP4	00562	CALL LOD16		
F3BF	2804	00563	JRZ LOD51-#		
F3C1	CB7A	00564	BIT 7, D		
F3C3	20C3	00565	JRNZ LOD0-#		
F3C5	CB7A	00566	BIT 7, D		
F3C7	2829	00567	JRZ LOD6-#		
F3C9	CD74F1	00568	CALL PRINT		
F3CC	1E	00569	DB NL		
F3CD		00570	DB 'FILE FOUND		
F3D8	00	00571	DB 0		
F3D9	PDE5	00572	PUSH IY		
F3DB	C1	00573	POP BC		
F3DC	2A0APP	00574	LD HL, (ARG1)		
F3DF	09	00575	ADD HL, BC		
F3E0	220APP	00576	LD (ARG1), HL		
F3E3	2A0CFF	00577	LD HL, (ARG2)		
F3E6	09	00578	ADD HL, BC		
F3E7	220CFF	00579	LD (ARG2), HL		
F3EA	78	00580	LD A, B		
F3EB	B1	00581	OR C		
F3EC	2804	00582	JRZ LOD6-#		
F3EE	AF	00583	XOR A		
F3EF	3208FF	00584	LD (FITY), A		
F3F2	DD2110FF	00585	; DATENBLOECKE LESEN		
F3F6	2A0APP	00586	LOD6: LD IX, DATA		
F3F9	CDADP4	00587	LD HL, (ARG1)		
F3FC	FE04	00588	LOD7: CALL LOD14		
		00589	CMP EOT		

;JA
 ;2.LESEN, KOPF UEBERLESEN
 ;BLOCK LESEN
 ;KOPF RICHTIG

 ;2.LESEN

 ;OFFSET

 ;OFFSET=0

 ;AUTOSTART UNTERDRUECKEN

 ;ANFANGSADR

F3FE	C8CF4	00590	JPZ	LOD12	
F401	FE03	00591	CMP	ETX	;ENDE?
F403	285E	00592	JRZ	LOD10-#	;JA
F405	FE1D	00593	CMP	GS	
F407	20F0	00594	JRNZ	LOD7-#	;KEIN RICHTIGER BLOCKANFANG
F409	OE00	00595	LD	C,0	
F40B	CDC8FF	00596	CALL	CRI	;BLOCKLAENGE
F40E	47	00597	LD	B,A	
F40F	4F	00598	LD	C,A	
F410	CBAA	00599	RES	5,D	
F412	CB7A	00600	BIT	7,D	
F414	2026	00601	JRNZ	LOD8-#	
F416	CBF2	00602	SET	6,D	;1.LESEN
F418	CB62	00603	BIT	4,D	;FUER UBERLESEN,2.LESEN
F41A	2020	00604	JRNZ	LOD8-#	
F41C	D5	00605	PUSH	DE	;ALLES KORR.
F41D	7A	00606	LD	A,D	
F41E	E60F	00607	AND	OFH	
F420	DD5601	00608	LD	D,(IX+1)	
		00609	;WAR BLOCK SCHON FALSCH		
F423	BA	00610	CMP	D	
F424	D1	00611	POP	DE	
F425	2015	00612	JRNZ	LOD8-#	;NEIN
F427	DD7E00	00613	LD	A,(IX)	
F42A	BB	00614	CMP	E	
F42B	200F	00615	JRNZ	LOD8-#	;NEIN
F42D	DD23	00616	INC	IX	
F42F	DD23	00617	INC	IX	
F431	D9	00618	EXX		
F432	15	00619	DEC	D	;KORREKTURZAehler-1

P433	D9	00620	EXX	JRNZ	LOD71-#		
P434	2002	00621		SET	4,D		
P436	CB82	00622		SET	6,D		
P438	CB82	00623	LOD71:	RES	5,D		
P43A	CBEA	00624		SET	5,D		
P43C	CDBFP4	00625	LOD8:	CALL	LOD16		
P43F	281F	00626		JRZ	LOD9-#		
P441	CB6A	00627		BIT	5,D		
P443	2047	00628		JRNZ	LOD12-#		
P445	CB7A	00629		BIT	7,D		
P447	2817	00630		JRZ	LOD9-#		
P449	D9	00631		EXX			
P44A	14	00632		INC	D		
P44B	7A	00633		LD	A,D		
P44C	D9	00634		EXX			
P44D	FE08	00635		CMP	8		
P44F	CA88F3	00636		JPZ	LOD0		
P452	DD7201	00637		LD	(IX+1),D		
P455	DDCB01BE	00638		RES	7,(IX+1)		
P459	DD7300	00639		LD	(IX+0),E		
P45C	DB23	00640		INC	IX		
P45E	DD23	00641		INC	IX		
P460	13	00642	LOD9:	INC	DE		
P461	1896	00643		JR	LOD7-#		
		00644		;ETX BEHANDLUNG			
P463	7A	00645	LOD10:	LD	A,D		
P464	E680	00646		AND	80H		
P466	57	00647		LD	D,A		
P467	2B	00648		DEC	HL		
P468	ED4B0CFF	00649		LD	BC,(ARG2)		

```

; ALLES KORRIGIERT?
; KEINE FALSCHEN BLOECKE MEHR
; FREIGABE RAM SCHREIBEN
; KORREKTURFLAG
; BLOCK LESEN
; PRUEFSUMME OK
; WURDE BLOCK EBEN KORRIGIERT?
; JA, FILE NICHT LESBAR
; NEIN
; FEHLER IGNORIEREN

; KORREKTURZAEHLER+1

; ZU VIELE FEHLER, NOCHMAL VERSUCHEN
; BLOCKNR. ABLEGEN
; FLAG LOESCHEN
; ERRORPOINTER
; BLOCKZAEHLER+1

; ZAEHLER LOESCHEN, 1/2-FLAG NICHT
; ENDADR.-1

```


F46C	ED42	00650	SBC	HL, BC
F46E	C28AF3	00651	JPNZ	LOD1
F471	D9	00652	EXX	
F472	7A	00653	LD	A, D
F473	B7	00654	OR	A
F474	2805	00655	JRZ	LOD11-#
F476	D9	00656	EXX	
F477	55	00657	LD	D, L
F478	C393F3	00658	JMP	LOD2
F47B	CD9BF4	00659	CALL	LOD13
F47E	3A08FF	00660	LD	A, (PITY)
F481	FE50	00661	CMP	'P'
F483	C0	00662	RNZ	
F484	7D	00663	LD	A, L
F485	FE51	00664	CMP	'Q'
F487	C8	00665	RZ	
F488	2A0EFF	00666	LD	HL, (ARG3)
F48B	E9	00667	JMP	(HL)
F48C	D9	00668	EXX	
F48D	CD74F1	00669	CALL	PRINT
F490		00670	DB	' BAD FILE
F49A	00	00671	DB	O
F49B	B7	00672	LD	A, E
F49C	7B	00673	OR	A
F49D	C8	00674	RZ	
F49E	CD31F1	00675	CALL	OUTH
F4A1	CD74F1	00676	CALL	PRINT
F4A4		00677	DB	' ERRORS'
F4AB	00	00678	DB	O
F4AC	C9	00679	RET	

F4AD	CDC8FF	00680	;SYNCHRONISIEREN
F4B0	3C	00681	LOD14: CALL CRI
F4B1	20FA	00682	INC A
F4B3	CDC8FF	00683	JRNZ LOD14-#
F4B6	FEFF	00684	LOD15: CALL CRI
F4B8	28F9	00685	CMP OFFH
F4BA	C9	00686	JRZ LOD15-#
		00687	RET
		00688	;BLOCK LESEN
F4BB	CDC8FF	00689	LOD16: CALL CRI
F4BE	CB72	00690	BIT 6,D
F4C0	2001	00691	JRNZ LOD17-#
F4C2	77	00692	LD (HL),A
F4C3	31	00693	LOD17: ADD C
F4C4	4F	00694	LD C,A
F4C5	23	00695	INC HL
F4C6	10F3	00696	DJNZ LOD16-#
F4C8	CDC8FF	00697	CALL CRI
F4CB	81	00698	ADD C
F4CC	CB82	00699	RES 6,D
F4CE	C8	00700	RZ
F4CF	D9	00701	EXX
F4D0	1C	00702	INC E
F4D1	D9	00703	EXX
F4D2	C9	00704	RET
		00705	*****
		00706	;*
		00707	SAVE
		00708	*****
		00709	*****
			;UEBERLESEN
			;PRUEFSUMME
			;PRUEFSUMME OK
			;FEHLERZAHLER+1

F4D3	EDFF	00710	DA	HEAD	
F4D5		00711	DB	'SAVE'	
F4D9	00	00712	NOP		
F4DA	CDB3F0	00713	SAEXT:	CALL	GFINA
F4DD	CDD2F0	00714		CALL	GFITY
F4E0	CD00F1	00715		CALL	ASHEX
F4E3	DA59F0	00716		JPC	ERROR
F4E6	220AFF	00717		LD	(ARG1),HL
F4E9	CD00F1	00718		CALL	ASHEX
F4EC	DA59F0	00719		JPC	ERROR
F4EF	220CFE	00720		LD	(ARG2),HL
F4F2	CD00F1	00721		CALL	ASHEX
F4F5	3003	00722		JRNC	SAVE-#
F4F7	211EF0	00723		LD	HL,RSTAR
F4FA	220EFD	00724	SAVE:	LD	(ARG3),HL
F4FD	CDE1F0	00725		CALL	GCOM
F500	CD61FA	00726		CALL	CINIT
F503	2189F5	00727		LD	HL,SAVEE
F506	E5	00728		PUSH	HL
F507	211AF5	00729		LD	HL,SAVE0
F50A	E5	00730		PUSH	HL
F50B	E5	00731		PUSH	HL
F50C	0E1E	00732		LD	C,NL
F50E	3A08FF	00733		LD	A,(FITY)
F511	FE20	00734		CMP	' '
F513	2005	00735		JRNZ	SAVE0-#
F515	3E50	00736		LD	A,'P'
F517	3208FF	00737		LD	(FITY),A
F51A	0614	00738	SAVE0:	LD	B,20
F51C	3EFF	00739	SAVE1:	LD	A,OFFH

					;FILENAME HOLEN
					;FILETYP
					;STARTADRESSE
					;ENDADRESSE
					;EINTRITTSUNKT
					;FEHLT, DESHALB RSTAR
					;RETURN=>SAVE0
					;FILEANFANG
					;KEINE-ANGABE
					;SYNCHRONZEICHEN

F51E	CDEFF	00740	CALL CPO		
F521	10F9	00741	DJNZ SAVE1-#		
F523	79	00742	LD A,C		
F524	CDEFF	00743	CALL CPO		
F527	3E01	00744	LD A,SOH		
F529	CDEFF	00745	CALL CPO		
		00746	;KOPF AUSGEBEN		
		00747	;SOH/FILENAME/TT/AA/EE/SS/...../PS/STX		
F52G	2100FF	00748	LD ' HL,FINA		
F52F	0620	00749	LD B,32		
F531	0E00	00750	LD C,0		
F533	7E	00751	LD A,(HL)		
F534	CDEFF	00752	CALL CPO		
F537	7E	00753	LD A,(HL)		
F538	81	00754	ADD C		
F539	4F	00755	LD C,A		
F53A	23	00756	INC HL		
F53B	10F6	00757	DJNZ SAVE2-#		
F53D	ED44	00758	NEG		
F53F	CDEFF	00759	CALL CPO		
F542	3E02	00760	LD A,STX		
F544	CDEFF	00761	CALL CPO		
F547	ED5E0AFF	00762	LD DE,(ARG1)		
F54B	0606	00763	SAVE3: LD B,6		
F54D	3EFF	00764	SAVE4: LD A,OPFH		
F54F	CDEFF	00765	CALL CPO		
F552	10F9	00766	DJNZ SAVE4-#		
F554	2A0CFF	00767	LD HL,(ARG2)		
F557	23	00768	INC HL		
F558	ED52	00769	SBC HL,DE		

;NL ODER SYNCHRONZEICHEN
;START KOPF

;PRUEFSUMME

;PUEFSUMME
;KOPF-ENDE
;ANF.ADR.
;SYNC.
;ENDADR.
;BLOCKLAENGE BERECHNEN

F55A	2826	00770	JRZ	SAVE8-#		
F55C	AF	00771	XOR	A		;PERTIG
F55D	B4	00772	OR	H		
F55E	2004	00773	JRNZ	SAVE5-#		
F560	85	00774	ADD	L		
F561	F266F5	00775	JPP	SAVE6		
F564	2E80	00776	LD	L,80H		
F566	45	00777	LD	B,L		;BYTEZAEHLER
F567	3E1D	00778	LD	A,GS		;START BLOCK
F569	CD8E9F	00779	CALL	CPO		
F56C	78	00780	LD	A,B		
F56D	4F	00781	LD	C,A		
F56E	CD8E9F	00782	CALL	CPO		;BLOCKLAENGE
F571	1A	00783	LD	A,(DE)		
F572	CD8E9F	00784	CALL	CPO		
F575	1A	00785	LD	A,(DE)		
F576	13	00786	INC	DE		
F577	81	00787	ADD	C		
F578	4F	00788	LD	C,A		
F579	10F6	00789	DJNZ	SAVE7-#		
F57B	ED44	00790	NEG			;PRUEFSUMME
F57D	CD8E9F	00791	CALL	CPO		
F580	1809	00792	JR	SAVE3-#		
F582	3E03	00793	LD	A,ETX		
F584	0EFF	00794	LD	C,OFFH		
F586	C38E9F	00795	JMP	CPO		;2.AUFZEICHNUNG
F589	0620	00796	LD	B,32		
F58B	3EFF	00797	LD	A,OFFH		
F58D	CD8E9F	00798	CALL	CPO		
F590	10F9	00799	DJNZ	SAVEF-#		

F592	3E04	LD	A,EOT	
F594	C3BEFF	JMP	CP0	

				* IN OUT

F597	EDFF	DA	HEAD	
F599		DB	'IN'	
F59B	00	NOP		
F59C	CD00F1	CALL	ASHEX	
F59F	4D	LD	C,L	
F5A0	ED78	IN	A	
F5A2	C331F1	JMP	OUTH	
F5A5	EDFF	DA	HEAD	
F5A7		DB	'OUT'	
F5AA	00	NOP		
F5AB	CD00F1	CALL	ASHEX	
F5AE	4D	LD	C,L	
F5AF	CD00F1	CALL	ASHEX	
F5B2	ED69	OUT	L	
F5B4	C9	RET		

				* FILL

F5B5	EDFF	DA	HEAD	

;PORTADRESSE HOLEN
;WERT HOLEN

P5B7		00830	DB	'FILL'	
P5B8	00	00831	NOP		
P5B9	CD00P1	00832	CALL	ASHEX	
P5BA	DA59F0	00833	JPC	ERROR	
P5BB	220AFF	00834	LD	(ARG1),HL	
P5BC	CD00P1	00835	CALL	ASHEX	
P5BD	DA59F0	00836	JPC	ERROR	
P5BE	220CFF	00837	LD	(ARG2),HL	
P5BF	CD00P1	00838	CALL	ASHEX	
P5B0	4D	00839	LD	C,L	
P5B1	2A0AFF	00840	LD	HL,(ARG1)	
P5B2	ED5E0CFF	00841	LD	DE,(ARG2)	
P5B3	3002	00842	JRNC	FILL1-#	
P5B4	0E00	00843	LD	C,0	
P5B5	71	00844	FILL1:	LD (HL),C	
P5B6	E5	00845	PUSH	HL	
P5B7	AF	00846	XOR	A	
P5B8	EB	00847	EX	DE,HL	
P5B9	ED52	00848	SBC	HL,DE	
P5BA	44	00849	LD	B,H	
P5BB	74D	00850	LD	C,L	
P5BC	E1	00851	POP	HL	
P5BD	54	00852	LD	D,H	
P5BE	5D	00853	LD	E,L	
P5BF	13	00854	INC	DE	
P5B0	EDB0	00855	LDIR		
P5B1	C9	00856	RET		
		00857			
		00858			
		00859			

;FUELLBYTE

;FUELLBYTE 00 FESTLEGEN

;*****

```

00860      P5EC      EDFF
00861      P5EE      DB
00862      P5F2      00
00863      P5F3      CD74F4
00864      P5F6      00
00865      P5FB      2AB1FF
00866      P5FC      CD4AF1
00867      P5FF      CD74F4
00868      P602      CD74F4
00869      P605      00
00870      P60D      CD32FA
00871      P611      CD4AF1
00872      P614      22B1FF
00873      P617      CDF8F0
00874      P61A      DB
00875      P61B      22B1FF
00876      P61E      C9

00860      ;* MEMSIZE
00861      ;*****
00862      ;*****
00863      DA      HEAD
00864      DB      'SIZE'
00865      NOP
00866      CALL PRINT
00867      DB      'SIZE:'
00868      DB      0
00869      LD      HL,(MSIZE)
00870      CALL OUTHL
00871      CALL PRINT
00872      DB      ' RAWTOP:'
00873      DB      0
00874      CALL MEMCH
00875      CALL OUTHL
00876      LD      (MSIZE),HL
00877      CALL INHEX
00878      RC
00879      LD      (MSIZE),HL
00880      RET

00881
00882
00883      ;*****
00884      ;* TESTPROGRAMME
00885      ;*****
00886      ;*****
00887      ;NMI VEKTOR EINTRAGEN
00888      DA      HEAD
00889      DB      'NMI'

```

;RANTEST

P624	00	NOP		
P625	216600	LD	HL, 66H	
P628	36C3	LD	(HL), 0C3H	
P62A	23	INC	HL	; JMP
P62B	3631	LD	(HL), L(BREAK)	
P62D	23	INC	HL	
P62E	36F6	LD	(HL), H(BREAK)	
P630	C9	RET		
P631	CD55F6	BREAK:	CALL UPUSH	; BEI NMI
P634	E1	POP	HL	; PC
P635	ED73ABFF	LD	(USP), SP	
P639	22ADFF	LD	(UPC), HL	
P63C	31FEFE	LD	SP, STACK	
P63F	CD7AF1	CALL	PRINT	
P642		DB	'BREAK AT '	
P64B	00	DB	0	
P64C	CD4AF1	CALL	OUTHL	
P64F	216CF0	LD	HL, MAIN	
P652	E5	PUSH	HL	
P653	ED45	RETN		
P655	ED73APFF	; CPU AUSRAEUMEN		
P659	31A9FF	UPUSH:	LD (SSP), SP	
P65C	FDE5	LD	SP, HIF	
P65E	DDE5	PUSH	IX	
P660	D9	PUSH	IX	
P661	08	KXI		
P662	E5	KXAF		
		PUSH	HL	; ERST ZWEITREGISTER

```

F663 D5          PUSH DB
F664 C5          PUSH BC
F665 F5          PUSH AF
F666 D9          EXX
F667 08          XIAF
F668 E5          PUSH HL
F669 D5          PUSH DE
F66A C5          PUSH BC
F66B F5          PUSH AF
F66C ED57        LD A,I
F66E 32A9FF       LD (HIF),A
F671 181C        JR UPOP1-#

00920          PUSH DB
00921          PUSH BC
00922          PUSH AF
00923          EXX
00924          XIAF
00925          PUSH HL
00926          PUSH DE
00927          PUSH BC
00928          PUSH AF
00929          LD A,I
00930          LD (HIF),A
00931          JR UPOP1-#
00932

; CPU EINRAEUMEN
00933          UPOP: LD (SSP),SP
00934          LD SP,UREG
00935          LD A,(HIF)
00936          LD I,A
00937          LD I,A
00938          POP AF
00939          POP BC
00940          POP DE
00941          POP HL
00942          EXX
00943          XIAF
00944          POP AF
00945          POP BC
00946          POP DE
00947          POP HL
00948          POP IX
00949          POP IY

```

;DANN HAUPTREGISTER

F6C1	36FF	00980	LD	(HL), OFFH		
F6C3	213800	00981	LD	HL, 38H		
F6C6	36C3	00982	LD	(HL), 0C3H		
F6C8	23	00983	INC	HL		
F6C9	36FC	00984	LD	(HL), L(BRK)		
F6CB	23	00985	INC	HL		
F6CC	36P6	00986	LD	(HL), H(BRK)		
F6CE	E1	00987	POP	HL		
F6CF	31C0FE	00988	LD	SP, USTCK		
F6D2	116CFO	00989	LD	DE, MAIN		
F6D5	D5	00990	PUSH	DE		
F6D6	E5	00991	PUSH	HL		
F6D7	189A	00992	JR	UPOP-#		
F6D9	FE50	00993	CMF	'P'		
F6DE	2011	00994	JRNZ	G03-#		
F6DD	13	00995	INC	DE		
F6DE	1A	00996	LD	A, (DE)		
F6DF	FE43	00997	CMF	'C'		
F6E1	C259F0	00998	JPNZ	ERROR		
F6E4	2AADFF	00999	LD	HL, (UPC)		
F6E7	ED7BABFF	01000	LD	SP, (USP)		
F6EB	E5	01001	PUSH	HL		
F6EC	1885	01002	JR	UPOP-#		
F6EE	2ACEFF	01003	LD	HL, (BRKPT)		
F6F1	7E	01004	LD	A, (HL)		
F6F2	FEFF	01005	CMF	OFFH		
F6F4	C259F0	01006	JPNZ	ERROR		
F6F7	3AD0FF	01007	LD	A, (BRKPT+2)		
F6FA	77	01008	LD	(HL), A		
F6FB	C9	01009	RET			

; RST38 EINTRAGEN
 ; JMP

 ; USERSTACK
 ; RUECKSPRUNG

 ; SPRUNG ADR.

F76C	CD55F6	01010							
F76F	E1	01011	BRK:	CALL	UPUSH				
F700	2B	01012		POP	HL				
F701	ED73ABFF	01013		DEC	HL				;ALTER PC
F705	22ADFF	01014		LD	(USP),SP				
F708	34FEFE	01015		LD	(UPC),HL				
F70B	3AD0FF	01016		LD	SP,STACK				
F70E	77	01017		LD	A,(BRKPT+2)				;BYTE WIEDER EINTRAGEN
F70F	2100F0	01018		LD	(HL),A				
F712	22CEFF	01019		LD	HL,START				
F715	216CF0	01020		LD	(BRKPT),HL				;BREAKPOINT LOESCHEN
F718	E5	01021		LD	HL,MAIN				
		01022		PUSH	HL				
		01023							
		01024							
F719	CD74F1	01025	REGDP:	CALL	PRINT				
F71C		01026		DB	,AF	BC	DE	HL	AF
F733	27	01027		DB	27H				
F734		01028		DB	,BC				
F738	27	01029		DB	27H				
F739		01030		DB	,DE				
F73D	27	01031		DB	27H				
F73E		01032		DB	,HL				
F742	27	01033		DB	27H				
F743		01034		DB	,IX	IY	I	SP	PC
F75B	1E	01035		DB	NL				
F75C	00	01036		DB	0				
F75D	060A	01037		LD	B,10				
F75F	DD2195FF	01038		LD	IX,UREG				
F763	DD6E00	01039	REGD1:	LD	L,(IX)				

F766	DD6601	01040	LD	H, (IX+1)	
F769	CD4AF1	01041	CALL	OUTH	
F76C	3E20	01042	LD	A, '	
F76E	CD02FF	01043	CALL	CO	
F771	DD23	01044	INC	IX	
F773	DD23	01045	INC	IX	
F775	10EC	01046	DJNZ	REGD1-#	
F777	3AA9FF	01047	LD	A, (HIF)	
F77A	CD31F1	01048	CALL	OUTH	
F77D	CD74F1	01049	CALL	PRINT	
F780	00	01050	DB	,	
F782	00	01051	DB	0	
F783	2AABFF	01052	LD	HL, (USP)	
F786	CD4AF1	01053	CALL	OUTH	
F789	CD74F1	01054	CALL	PRINT	
F78C	00	01055	DB	,	
F78E	00	01056	DB	0	
F78F	2AADFF	01057	LD	HL, (UPC)	
F792	CD4AF1	01058	CALL	OUTH	
F795	C9	01059	RET		
F796	C319F7	01060	REGD:	JMP REGDF	
F799	EDFF	01061	DA	HEAD	
F79B	00	01062	DB	'REG'	
F79E	00	01063	NOP		
F79F	CD55F1	01064	CALL	FNEXT	
F7A2	3BF2	01065	JRC	REGD-#	
F7A4	2109F8	01066	LD	HL, RTAB	
F7A7	010800	01067	LD	BC, 8	
F7AA	EDB1	01068	LD	BC, 8	
		01069	CP	IR	

;WEIL CR

;AFBCDEHL SUCHEN

F7AC	201A	01070	JRNZ	REG4-#	; DOPPELREG.
F7AE	018B07	01071	LD	BC,UREG-RTAB-1	
F7B1	09	01072	ADD	HL,BC	; ADRESSE DES REG.
F7B2	13	01073	INC	DE	
F7B3	1A	01074	LD	A,(DE)	
F7B4	FE27	01075	CMP	27H	; STRICHREGISTER
F7B6	2005	01076	JRNZ	REG2-#	; NEIN
F7B8	13	01077	INC	DE	
F7B9	3E08	01078	LD	A,8	; WEIL STRICHREGISTER
F7BB	85	01079	ADD	L	; ADR KORRIGIEREN
F7BC	6F	01080	LD	L,A	
F7BD	E5	01081	PUSH	HL	
F7BE	CD00F1	01082	CALL	ASHEX	; BYTE HOLEN
F7C1	7D	01083	LD	A,L	
F7C2	E1	01084	POP	HL	
F7C3	38D1	01085	JRC	REGD-#	; KEIN HEX
F7C5	77	01086	LD	(HL),A	; EINTRAGEN
F7C6	18D7	01087	JR	REG1-#	
F7C8	FE49	01088	CMP	I,	; DOPPELREGISTER
F7CA	2014	01089	JRNZ	REG5-#	
F7CC	13	01090	INC	DE	
F7CD	1A	01091	LD	A,(DE)	
F7CE	FE58	01092	CMP	X,	; IX REGISTER
F7D0	21A5FF	01093	LD	HL,HIX	
F7D3	2825	01094	JRZ	REG7-#	
F7D5	FE59	01095	CMP	Y,	
F7D7	23	01096	INC	HL	
F7D8	23	01097	INC	HL	
F7D9	281F	01098	JRZ	REG7-#	; IY REGISTER
F7DB	21A9FF	01099	LD	HL,HIF	; INTERRUPTREGISTER

F7DE	18DD	01100	REG5:	JR	REG2-#	
F7E0	FE50	01101		CMP	'P'	
F7E2	200B	01102		JRNZ	REG6-#	
F7E4	13	01103		INC	DE	
F7E5	1A	01104		LD	A,(DE)	
F7E6	FE43	01105		CMP	'C'	
F7E8	20AC	01106		JRNZ	REGD-#	
F7EA	21ADFF	01107		LD	HL,UPC	
F7ED	180B	01108		JR	REG7-#	
F7EF	FE53	01109	REG6:	CMP	'S'	
F7F1	13	01110		INC	DE	
F7F2	1A	01111		LD	A,(DE)	
F7F3	FE50	01112		CMP	'P'	
F7F5	209F	01113		JRNZ	REGD-#	
F7F7	21ABFF	01114		LD	HL,USP	
F7FA	E5	01115	REG7:	PUSH	HL	
F7FB	13	01116		INC	DE	
F7FC	CD00F1	01117		CALL	ASHEX	
F7FF	44	01118		LD	B,H	
F800	4D	01119		LD	C,L	
F801	E1	01120		POP	HL	
F802	3892	01121		JRC	REGD-#	
F804	71	01122		LD	(HL),C	
F805	23	01123		INC	HL	
F806	70	01124		LD	(HL),B	
F807	1896	01125		JR	REG1-#	
F809		01126	RTAB:	DB	'FACEDLH'	
		01127				
		01128				
		01129				

; FALSCH EINGABE
; PC

; FALSCH EINGABE
; STACKPOINTER

; WORT HOLEN

; FALLS KEINE HEX-EINGABE
; EINTRAGEN


```

01130
01131
01132
01133
01134
01135
01136
01137
01138
01139
01140
01141
01142
01143
01144
01145
01146
01147
01148
01149
01150
01151
01152
01153
01154
01155
01156
01157
01158
01159

F811 EDFF
F813
F817 00
F818 210000
F81B 010000
F81E 3E1E
F820 CDC2FF
F823 CD74F4
F826
F829 00
F82A 3EED
F82C EDB1
F82E C0
F82F 3EFF
F831 BE
F832 20F6
F834 E5
F835 C5
F836 0610
F838 7E
F839 23
F83A B7
F83B 2806
F83D 10F9
F83F C1
F840 E1

      DA HEAD
      DB 'HELP',
      NOP
      LD HL,0
      LD BC,0
      LD A,NL
      CALL C0
      HELP1: CALL PRINT
             DB ', '
             DB 0
      HELP2: LD A,L(HEAD)
             CPIR
             RNZ
             LD A,H(HEAD)
             CMP (HL)
             JRNZ HELP2-#
             PUSH HL
             PUSH BC
             LD B,16
             DBLP3: LD A,(HL)
                    INC HL
                    OR A
             JRZ HELP4-#
             DJNZ HELP3-#
             POP BC
             POP HL

```

;SUCHE NACH ED

;FOLGT FF?

;NEIN

;TEST, OB NACH SPÄTESTENS
;16 ZEICHEN EIN NOP FOLGT

;JA, AUSGABE

```

F841 18E7 JR      HELP2-#
F843 C1  HELP4: POP BC
F844 E1  POP HL
F845 23  HELP5: INC HL
F846 0B  DEC BC
F847 7E  LD  A,(HL)
F848 B7  OR  A
F849 2B8 JRZ  HELP1-#
F84B CDC2FF CALL CO
F84E 1BF5 JR   HELP5-#

01160 *****
01161 ; * READMC
01162 ; *****
01163 ; *
01164 ; *****
01165 ; *
01166 ; *****
01167 ; *
01168 ; *****
01169 ; *
01170 ; *****
01171 ; *
01172 ; *****
01173 ; *
01174 ; *****
01175 ; *
01176 ; *****
01177 ; *
01178 ; *****
01179 ; *
01180 ; *****
01181 ; *
01182 ; *****
01183 ; *
01184 ; *****
01185 ; *
01186 ; *****
01187 ; *
01188 ; *****
01189 ; *

F850 EDFF DA  HEAD
F852 F852 DB  'READ'
F856 00  NOP
F857 CD7BFA CALL RINIT
F85A CDD2F0 CALL GFITY
F85D CD00F1 CALL ASHEX
F860 110000 LD  DE,0
F863 3801 JRC  READ1-#
F865 EB  EX  DE,HL
F866 CDC5FF READ1: CALL RI
F86E FE3A  CMP 3AH
F86B 20F9 JRNZ READ1-#
F86D 0E00 LD  C,0

;AUSGABE

;TEST ENDE

;DOPPELPUNKT SUCHEN

;PRUEFSUMME =0 SETZEN

```

F86F	CDCEP8	01190	CALL READ3		; BLOCKLANGE
F872	B7	01191	OR A		
F873	C8	01192	RZ		; ENDE?
F874	47	01193	LD B,A		; FERTIG
F875	CDCEP8	01194	CALL READ3		
F878	67	01195	LD H,A		; HI DER ADR.
F879	CDCEP8	01196	CALL READ3		
F87C	6F	01197	LD L,A		; LO
F87D	3A08FF	01198	LD A,(FITY)		
F880	FE41	01199	CMP A'		
F882	2017	01200	JNZ REA12-#		
F884	C5	01201	PUSH BC		
F885	D5	01202	PUSH DE		
F886	E5	01203	PUSH HL		
F887	CD4AF1	01204	CALL OUTHL		
F88A	3E20	01205	LD A,' '		
F88C	3208FF	01206	LD (FITY),A		
F88F	CDP8F0	01207	CALL INHEX		
F892	D1	01208	POP DE		
F893	3803	01209	JRC REA11-#		
F895	ED52	01210	SBC HL,DE		
F897	E3	01211	EX (SP),HL		
F898	E1	01212	REA11: POP HL		
F899	EB	01213	EX DE,HL		
F89A	C1	01214	POP BC		; ADR+OFFSET
F89B	19	01215	ADD HL,DE		
F89C	CDCEP8	01216	CALL READ3		
F89F	CDCEP8	01217	CALL READ3		; DATEN
F8A2	77	01218	LD (HL),A		
		01219			

P8A3	23	01220	INC HL	
P8A4	10F9	01221	DJNZ READ2-#	
P8A6	CDCEFF8	01221	CALL READ3	
P8A9	28BE	01222	JRZ READ1-#	
P8AB	CD74F1	01224	CALL PRINT	
P8AE		01225	DB 'CHECKSUM ERROR	
P8BD	00	01226	DB 0	
P8BE	CD4AF1	01227	CALL OUTHL	
P8C1	CDCEFF7	01228	CALL CI	
P8C4	FE51	01229	CMP 'Q'	
P8C5	08	01230	RZ	
P8C7	3E1E	01231	LD A,NL	
P8C9	CD02FF	01232	CALL CO	
P8CC	1898	01233	JR READ1-#	
P8CE	D5	01234	PUSH DE	
P8CF	CD05FF	01235	CALL RI	
P8D2	CD20F1	01236	CALL CNVEM	
P8D5	07	01237	RLCA	
P8D6	07	01238	RLCA	
P8D7	07	01239	RLCA	
P8D8	07	01240	RLCA	
P8D9	57	01241	LD D,A	
P8DA	CD05FF	01242	CALL RI	
P8DD	CD20F1	01243	CALL CNVEM	
P8E0	E2	01244	OR D	
P8E1	57	01245	LD D,A	
P8E2	81	01246	ADD C	
P8E3	4F	01247	LD C,A	
P8E4	7A	01248	LD A,D	
P8E5	D1	01249	POP DE	

;PRUEFSUMME
;OK

;HALT BIS TASTE GEDRUECKT
;QUIT?
;JA

;BYTE VOM LESER

;BYTE ZUSAMMENFUEGEN
;RETTEN
;PRUEFSUMME
;NACH C ZURUECK

READ3:

P8B6	C9	01250	RET	
		01251		*****
		01252		*****
		01253	;	*****
		01254	;	*****
		01255	;	*****
		01256	DA	HEAD
P8E7	EDFF	01257	DB	'CHECKSUM'
P8E9	00	01258	NOP	
P8F1	00	01259	CALL	ASHEX
P8F2	CD00F1	01260	JPC	ERROR
P8F5	DA59F0	01261	PUSH	HL
P8F8	E5	01262	CALL	ASHEX
P8F9	CD00F1	01263	EX	DE,HL
P8FC	EB	01264	INC	DE
P8FD	13	01265	POP	HL
P8FE	E1	01266	JPC	ERROR
P8FF	DA59F0	01267	LD	IX,0
P902	DD210000	01268	LD	B,0
P906	0600	01269	LD	C,(HL)
P908	4E	01270	ADD	IX,BC
P909	DD09	01271	INC	HL
P90B	23	01272	PUSH	HL
P90C	E5	01273	XOR	A
P90D	AF	01274	SBC	HL,DE
P90E	ED52	01275	POP	HL
P910	E1	01276	JRC	CHK1-#
P911	38F5	01277	PUSH	IX
P913	DD55	01278	POP	HL
P915	E1	01279	JMP	OUTHL
P916	C34AF1			

;ENDADRESSE+1

;STARTADR.

;IX DIENST ALS AKKU

;SUMMIEREN

;CY=0

;ENDADRESSE ERREICHT?

;NEIN,WEITER

;ERGEBNIS

;NACH HL FUER AUSGABE

;AUSGABE UND RETURN

```

01280
01281
01282
01283
01284
01285
01286
01287
01288
01289
01290
01291
01292
01293
01294
01295
01296
01297
01298
01299
01300
01301
01302
01303
01304
01305
01306
01307
01308
01309

F919 EDFF
F91B
F91C 00
F91D CD00F1
F920 E5
F921 E5
F922 E5
F923 CD00F1
F926 EB
F927 E1
F928 19
F929 CD74F1
F92C
F930 00
F931 CD4AF1
F934 E1
F935 B7
F936 ED52
F938 CD74F1
F93B
F941 00
F942 CD4AF1
F945 E1
F946 EB
F947 B7

DA HEAD
DB ' ',
NOP
CALL ASHEX
PUSH HL
PUSH HL
PUSH HL
CALL ASHEX
EX DE,HL
POP HL
ADD HL,DE
CALL PRINT
DB 'SUM:',
DB 0
CALL OUTHL
POP HL
OR A
SBC HL,DE
CALL PRINT
DB ' DIF:',
DB 0
CALL OUTHL
POP HL
EX DE,HL
OR A

;*****
;* CALCULATE *
;*****

; 1.WERT
; SUMME AUSRECHNEN
; UND AUSGEBEN

; DIFFERENZ AUSRECHNEN
; CY=0
; UND AUSGEBEN

```

F948	ED52	01310	SBC	HL,DE	
F94A	2B	01311	DEC	HL	
F94B	2B	01312	DEC	HL	
F94C	CD74F1	01313	CALL	PRINT	
F94F		01314	DB	' REL:	
F955	00	01315	DB	0	
F956	29	01316	ADD	HL,HL	
F957	7C	01317	LD	A,H	
F958	CE00	01318	ADC	0	
F95A	2005	01319	JRNZ	CALC1-#	
F95C	7D	01320	LD	A,L	
F95D	1F	01321	RRA		
F95E	C331F1	01322	JMP	OUTH	
F961	CD74F1	01323	CALC1:	CALL PRINT	
F964		01324	DB	'**'	
F966	00	01325	DB	0	
F967	C9	01326	RET		
		01327			
		01328	*****		
		01329	;* FIND		*
		01330	*****		
		01331	*****		
F968	EDFF	01332	DA	HEAD	
F96A		01333	DB	'FIND'	
F96E	00	01334	NOP		
F96F	CDD2F0	01335	FIND:	CALL GPITY	
F972	CD00F1	01336		CALL ASHEX	
F975	E5	01337		PUSH HL	
F976	CD00F1	01338		CALL ASHEX	
F979	CD55F1	01339		CALL FNEXT	


```

;TEST BEREICHS UEBERSCHREITUNG
;BEI RELATIVEN SPRUENGEN
;ZU WEIT
;/2
;AKKU AUSGEBEN UND RETURN
;WENN ZU WEIT

```

F97C	ED5340FF	01340	LD	(TXTPT),DE			
F980	E5	01341	PUSH	HL			
F981	3A08FF	01342	LD	A, (FITY)			
F984	21BFF9	01343	LD	HL, FINCH			
F987	1110FF	01344	LD	DE, DATA			
F98A	010C00	01345	LD	BC, 12			
F98D	FE41	01346	CMP	'A'			
F98F	2803	01347	JRZ	FIND1-#			
F991	21C9F9	01348	LD	HL, FINHX			
F994	EDB0	01349	FIND1:	LDIR			
F996	C1	01350	POP	BC			
F997	03	01351	INC	BC			
F998	E1	01352	POP	HL			
F999	ED5B40FF	01353	FIND2:	LD DE, (TXTPT)			
F99D	E5	01354	PUSH	HL			
F99E	C310FF	01355	FIND3:	JMP DATA			
F9A1	BE	01356	FIND4:	CMP (HL)			
F9A2	23	01357	INC	HL			
F9A3	28F9	01358	JRZ	FIND3-#			
F9A5	79	01359	FIND7:	LD A, C			
F9A6	95	01360	SUB	L			
F9A7	78	01361	LD	A, B			
F9A8	9C	01362	SBC	H			
F9A9	E1	01363	POP	HL			
F9AA	3010	01364	JRNC	FIND6-#			
F9AC	C9	01365	RET				
F9AD	1A	01366	FIND8:	LD A, (DE)			
F9AE	FB0D	01367	CMP	CR			
F9B0	20F3	01368	JRNZ	FIND7-#			
F9B2	E1	01369	FIND5:	POP HL			

F9B3	CD4AF1	01370	CALL OUTHL		
F9B6	CD74F1	01371	CALL PRINT		
F9B9		01372	DB ', ',		
F9BB	00	01373	DB 0		
F9BC	23	01374	INC HL		
F9BD	18DA	01375	JR FIND2-#		
		01376			
F9BF	1A	01377	FINCH: LD A, (DE)		
F9C0	13	01378	INC DE		
F9C1	FE0D	01379	CMF CR		
F9C3	C2A1F9	01380	JPNZ FIND4		
F9C6	C3B2F9	01381	JMP FIND5		
F9C9	E5	01382	FINHX: PUSH HL		
F9CA	CD00F1	01383	CALL ASHEX		
F9CD	7D	01384	LD A, L		
F9CE	E1	01385	POP HL		
F9CF	D2A1F9	01386	JPMC FIND4		
F9D2	C3ADF9	01387	JMP FIND8		
		01388			
		01389	*****		
		01390	; * EOF		
		01391	; *****		
		01392			
F9D5	EDFF	01393	DA HEAD		
F9D7		01394	DB 'EOF'		
F9DA	00	01395	NOP		
F9DB	3AB3FF	01396	BOFI: LD A, (EOPCH)		
F9DE	CD31F1	01397	CALL OUTH		
F9E1	CD8F0	01398	CALL INHEX		
F9E4	D8	01399	RC		

; ADRESSE AUSGEBEN
 ; WEITERSUCHEN
 ; STRING SUCHEN
 ; VERGLEICHEN
 ; GEFUNDEN
 ; HEX SUCHEN
 ; VERGLEICHEN
 ; GEFUNDEN ??

F9E5	7D	LD	A,L	
F9E6	32B3FF	LD	(EOFCH),A	
F9E9	C9	RET		

		;* POWER OFF		*
		*****		*
F9EA	EDFF	DA	HEAD	
F9EC		DB	'OFF'	
F9EF	00	NOP		
F9F0	D300	OFF:	OUT 0	
F9F2	D300	OUT	0	
F9F4	C9	RET		

		;* ASNRI		*
		*****		*
F9F5	EDFF	DA	HEAD	
F9F7		DB	'ASNRI=RI'	
F9FF	00	NOP		
FA00	218CFA	RIRI:	LD HL,RI1	
FA03	180F	JR	ASNRI-#	
FA05	EDFF	DA	HEAD	
FA07		DB	'ASNRI=CRI'	
FA10	00	NOP		
FA11	21A3FA	RICRI:	LD HL,CRI1	

FA14	22C6FF	01430	ASNRI: LD	(RI+1),HL	
FA17	C9	01431	RET		
		01432			
FA18	EDFF	01433	DA	HEAD	
FA1A		01434	DB	'ASNRI=USR'	
FA23	00	01435	NOP		
FA24	CD00F1	01436	CALL	ASHEX	
FA27	D8	01437	RC		
FA28	18EA	01438	JR	ASNRI-#	
		01439			
FA2A	3AB2FF	01440	MEMSI: LD	A,(MSIZB+1)	
FA2D	47	01441	LD	B,A	
FA2E	3AB1FF	01442	LD	A,(MSIZE)	
FA31	C9	01443	RET		
		01444			
		01445			
FA32	21FFFF	01446	:EIMPACHER RANTEST		
FA35	1100E8	01447	MEMCH: LD	HL,-1	
FA38	23	01448	LD	DE,SYSEB	
FA39	7E	01449	INC	HL	
FA3A	EEFF	01450	XOR	A,(HL)	
FA3C	77	01451	LD	(HL),A	
FA3D	BE	01452	CMP	(HL)	
FA3E	200C	01453	JRNZ	MCH1-#	
FA40	EEFF	01454	XOR	OFFH	
FA42	77	01455	LD	(HL),A	
FA43	BE	01456	CMP	(HL)	
FA44	2006	01457	JRNZ	MCH1-#	
FA46	E5	01458	PUSH	HL	
FA47	ED52	01459	SBC	HL,DE	

; WIRD VON EXTERNEN PGM GENUTZT

; START BEI 0

; BIS SYSTEMBEGINN

; KOMPLEMENTIEREN

; WIEDER GLEICH?

; NEIN

; ENDE?

FA63	0608	01490	LD	B,8
FA65	2173FA	01491	LD	HL,CTAB*
FA68	EDB3	01492	OTIR	
FA6A	3E05	01493	LD	A,5
FA6C	D316	01494	OUT	CTC2
FA6E	3E04	01495	LD	A,4
FA70	D316	01496	OUT	CTC2
FA72	C9	01497	RET	
		01498		
FA73	04	01499	CTAB:	DB 4
FA74	04	01500	DB	4
FA75	01	01501	DB	1
FA76	40	01502	DB	40H
FA77	03	01503	DB	3
FA78	C1	01504	DB	OC1H
FA79	05	01505	DB	5
FA7A	EA	01506	DB	OEAH
		01507		
		01508	;LOCHBANDLESEN INITIALISIEREN	
FA7B	3E4F	01509	RINIT:	LD A,04FH
FA7D	D30D	01510	OUT	CP1A
FA7F	3ECF	01511	LD	A,OCFH
FA81	D30F	01512	OUT	CP1B
FA83	3E01	01513	LD	A,1
FA85	D30F	01514	OUT	CP1B
FA87	3E10	01515	LD	A,10H
FA89	D30F	01516	OUT	CP1B
FA8B	C9	01517	RET	
		01518		
		01519	;ZEICHEN VON LOCHBAND HOLEN	

```

PASC 3E10          LD  A,10H          ;STARTIMPULS
PABE D30E          OUT DP1B
FA90 AF           XOR  A
FA91 D30E          OUT DP1B
FA93 CDB8FA       RI12: CALL SEOF
FA96 DB0E          IN  DP1B
FA98 E601          AND  1
FA9A 20F7          JRNZ RI12-#
FA9C DB0C          IN  DP1A
FA9E B67F          AND  7FH
FAA0 28EA          JRZ  RI1-#
FAA2 C9           RET

PAA3 CDB8FA       ;ZEICHEN VON KASSETTE HOLEN
PAA6 DB19          CRI1: CALL SEOF
PAA8 E601          IN  CS10A
PAAA 28F7          AND  1
FAAC DB18          JRZ  CRI1-#
FAAE C9           IN  DS10A
FAAF D318          RET
PAB1 DB19          ;ZEICHEN AUF KASSETTE AUFZEICHNEN
PAB3 E604          CP01: OUT DS10A
PAB5 28FA          CP011: IN  CS10A
PAB7 C9           AND  4
PAB8 DB0B          JRZ  CP011-#
PABA FE03          RET
PABC C0           SEOF: IN  KEYBD
                                CMP  EOF
                                RNZ

                                ;LESEN UNTERBRECHEN?

```

FABD F1	01550	POP AF		
FABE 3AB3FF	01551	LD A,(BOPCH)		
FAC1 C9	01552	RET		
	01553			
FAC2 DB09	01554	;CONSOLSTATUSABFRAGE		
FAC4 B7	01555	CSTS: IN KEYBC		
FAC5 C8	01556	OR A		
FAC6 DB08	01557	RZ		
FAC8 FE0D	01558	IN KEYBD		
FACA 28F6	01559	CMF CR		
FACC 32B4FF	01560	JRZ CSTS-#		
FACP F6FF	01561	LD (CSTS),A		
FAD1 C9	01562	OR OFFH		
	01563	RET		
	01564			
FAD2 3AB4FF	01565	;TASTATUREINGABE		
FAD5 B7	01566	CI1: LD A,(CSTS)		
FAD6 2807	01567	OR A		
FAD8 AF	01568	JRZ CI11-#		
FAD9 32B4FF	01569	XOR A		
FADC DB08	01570	LD (CSTS),A		
FAD8 C9	01571	IN KEYBD		
FADF DB09	01572	RET		
FAE1 B7	01573	CI11: IN KEYBC		
FAE2 20FB	01574	OR A		
FAE4 DB09	01575	JRNZ CI11-#		
FAE6 B7	01576	IN KEYBC		
FAE7 28FB	01577	OR A		
FAE9 DB08	01578	JRZ CI12-#		
	01579	IN KEYBD		

;STACK KORRIGIEREN

;KEINE TASTE

;AUF STEIGENDE FLANKE WARTEN

FB17	28EF	01610	JRZ	CRT4-#			
FB19	FB08	01611	CMP	CURL			; LINEFEED AUSFUEHREN
FB1B	380B	01612	JRC	CRT5-#			
FB1D	203A	01613	JRNZ	CRT10-#			
FB1F	2B	01614	DEC	HL			
FB20	7C	01615	LD	A, H			; CURSOR LINKS
FB21	FE8	01616	CMP	H(CRTBG)			; TEST OB NOCH IM BILDSCHIRM
FB23	D84FB	01617	JPC	CRT15			
FB26	1825	01618	JR	CRT9-#			
FB28	77	01619	LD	(HL), A			; ZEICHEN NACH CRT
FB29	23	01620	INC	HL			; CURSOR RECHTS
FB2A	7C	01621	LD	A, H			; ROLLEN NOTWENDIG?
FB2B	FE8	01622	CMP	H(CRTEN)			
FB2D	381E	01623	JRC	CRT9-#			; NEIN
FB2F	1100B8	01624	LD	DE, CRTBG			
FB32	E5	01625	PUSH	HL			
FB33	2140B8	01626	LD	HL, CRTBG+CHPL			
FB36	010006	01627	LD	BC, CRTEN-CRTBG			; ROLLEN
FB39	EDB0	01628	LDIR				
FB3B	21C0ED	01629	LD	HL, CRTEN-CHPL			
FB3E	0640	01630	LD	B, CHPL			
FB40	3E20	01631	LD	A, '			
FB42	77	01632	LD	(HL), A			; LETZTE ZEILE LOESCHEN
FB43	23	01633	INC	HL			
FB44	10FC	01634	DJNZ	CRT8-#			
FB46	E1	01635	POP	HL			; CURSOR KORRIGIEREN
FB47	114000	01636	LD	DE, CHPL			
FB4A	B7	01637	OR	A			
FB4B	ED52	01638	SBC	HL, DE			
FB4D	3E80	01639	LD	A, 80H			; CURSOR AN

FB4F	AE	01640	XOR	(HL)	
FB50	77	01641	LD	(HL),A	
FB51	2293FF	01642	LD	(CURS),HL	
FB54	F1	01643	POP	AF	
FB55	C1	01644	POP	BC	
FB56	D1	01645	POP	DE	
FB57	E1	01646	POP	HL	
FB58	C9	01647	RET		
FB59	FE09	01648	CRT10: CMP	CURR	
FB5B	28CC	01649	JRZ	CRT6-#	
FB5D	FE0B	01650	CMP	CURU	
FB5F	2005	01651	JRNZ	CRT11-#	
FB61	B7	01652	OR	A	
FB62	ED52	01653	SBC	HL,DE	
FB64	18BA	01654	JR	CRT4-#	
FB66	FE1B	01655	CMP	ESC	
FB68	2005	01656	JRNZ	CRT12-#	
FB6A	7E	01657	LD	A,(HL)	
FB6B	EE80	01658	XOR	80H	
FB6D	18B9	01659	JR	CRT5-#	
FB6F	FE0C	01660	CMP	CLS	
FB71	200D	01661	JRNZ	CRT14-#	
FB73	2100E8	01662	LD	HL,CRTBG	
FB76	3620	01663	CRT13: LD	(HL),'	
FB78	23	01664	INC	HL	
FB79	7C	01665	LD	A,H	
FB7A	FEFE	01666	CMP	H(CRTEN)	
FB7C	20F8	01667	JRNZ	CRT13-#	
FB7E	3E0E	01668	LD	A,HOME	
FB80	FE0E	01669	CRT14: CMP	HOME	

;CURSOR HOCH

;INVERTIEREN

;LOESCHEN

;ANSCHLIESSEND HOME

FB82	20A4	JRNZ CRT5-#	MSIZE
FB84	2100B8	CRT15: LD HL,CRTBG	EOFFCH
FB87	18C4	JR CRT9-#	CSTSR
01670			
01671			
01672			
01673			
01674			
01675			
01676			
01677			
01678			
01679			
01680			
01681			
01682			
01683			
01684			
01685			
01686			
01687			
01688			
01690			
01691			
01692			
01693			
01694			
01695			
01696			
01697			
01698			
01699			

FBB9	CFF4	01700	DA	MEM01	;8
FBBB	40F3	01701	DA	MOVE	;9
FBBD	63F8	01702	DA	READ1-3	;10 =0AH
FBBF	6EF3	01703	DA	LDEXT	;11 =0BH
FBC1	D4F4	01704	DA	SAEXT	;12 =0CH
FBC3	F0F9	01705	DA	OFF	;13 =0DH
FBC5	00FA	01706	DA	RIRI	;14 =0EH
FBC7	11FA	01707	DA	RICRI	;15 =0FH
FBC9	61FA	01708	DA	CINIT	;16 =10H
FBCB	7BFA	01709	DA	RINIT	;17 =11H
FBCD	DDF5	01710	DA	FILL1	;18 =12H
FBCF	25F6	01711	DA	NMIV	;19 =13H
FBD1	55F6	01712	DA	UPUSH	;20 =14H
FBD3	73F6	01713	DA	UPOP	;21 =15H
FBD5	19F7	01714	DA	REGDP	;22 =16H
FBD7	9FF7	01715	DA	REG1	;23 =17H
FBD9	02F9	01716	DA	CHECK	;24 =18H
FBD8	6FF9	01717	DA	FIND	;25 =19H
FBD8	32FA	01718	DA	MEMCH	;26 =1AH
		01719			
		01720			
		01721			
		01722	EJEC		
		01723	*****		
		01724	;* VEREINBARUNGEN		
		01725	*****		
		01726	*****		
		01727	PORTS:		
		01728	POWOF: EQU 0		;POWER OFF
		01729	KEYBD: EQU 8		;TASTATURDATEN

01730	KEYBC:	EQU	9	TASTATURSTATUS	OFFH=GEDR./0=KEINE TASTE
01731	BEP0F:	EQU	8	;SIGNALTON	AUS
01732	BEPON:	EQU	9	;AN	
01733	DP1A:	EQU	0CH	;12	PIO 1
01734	CP1A:	EQU	0DH	;13	
01735	DP1B:	EQU	0EH	;14	
01736	CP1B:	EQU	0FH	;15	
01737	DP2A:	EQU	10H	;16	STEUERPORT LBL
01738	CP2A:	EQU	11H	;17	BIT 0 STATUS,BIT 4 STARTIMPULS
01739	DP2B:	EQU	12H	;18	LBL DATEN
01740	CP2B:	EQU	13H	;19	
01741	CTC0:	EQU	14H	;20	
01742	CTC1:	EQU	15H	;21	
01743	CTC2:	EQU	16H	;22	KASSETTE
01744	CTC3:	EQU	17H	;23	
01745	DSIOA:	EQU	18H	;24	KASSETTE
01746	CSIOA:	EQU	19H	;25	
01747	DSIOB:	EQU	1AH	;26	
01748	CSIOB:	EQU	1BH	;27	
01749	;CURL:	EQU	8	;CURSOR	LINKS
01750	CURR:	EQU	9	;RECHTS	
01751	CURD:	EQU	0AH	;10	;RUNTER
01752	CURU:	EQU	0BH	;11	;HOCH
01753	HOME:	EQU	0EH	;14	
01754	ESC:	EQU	1BH	;ESCAPE	
01755	NL:	EQU	1EH	;NEW LINE	
01756	LF:	EQU	0AH	;LINE FEED	
01757	CR:	EQU	0DH	;RETURN	
01758	CLS:	EQU	0CH	;BILDSCHIRM	LOESCHEN
01759					

01760	SOH:	EQU	1
01761	GS;	EQU	1DH
01762	ETX:	EQU	03H
01763	EXT:	EQU	04H
01764	STX:	EQU	02H
01765	ERASE:	EQU	06H
01766	EOP:	EQU	ETX
01767	MEMB:	EQU	16
01768	MEML:	EQU	16
01769	CHPL:	EQU	64
01770	HEAD:	EQU	OFFEDH
01771			
01772		ORG	OFFOOH
01773		;RAM BEREICH	
01774	STACK:	EQU	#-2
01775	USTCK:	EQU	#-64
01776	FINA:	BER	8
01777	FITY:	BER	2
01778	ARG1:	BER	2
01779	ARG2:	BER	2
01780	ARG3:	BER	2
01781	DATA:	BER	16
01782	INTV:	BER	32
01783	TXTPT:	BER	2
01784	TXTBU:	BER	81
01785	BULEN:	EQU	81
01786	CURS:	BER	2
01787	HAF:	BER	2
01788	HBC:	BER	2
01789	HDE:	BER	2
FF00			
FF08			
FF0A			
FF0E			
FF10			
FF20			
FF40			
FF42			
FF93			
FF95			
FF97			
FF99			

PF9B	01790	HHL:	BER	2	
PF9D	01791	AAP:	BER	2	
PF9F	01792	ABC:	BER	2	
FFA1	01793	ADE:	BER	2	
FFA3	01794	AHL:	BER	2	
FFA5	01795	HIX:	BER	2	
FFA7	01796	HIY:	BER	2	
FFA9	01797	HIP:	BER	2	
	01798	UREG:	ERU	HAF	
FFAB	01799	USP:	BER	2	
FFAD	01800	UPC:	BER	2	
FFAF	01801	SSP:	BER	2	
FFB1	01802	MSIZE:	BER	2	
FFB3	01803	EOFCH:	BER	1	
FFB4	01804	CSTSR:	BER	1	
FFB5	01805	LOB:	BER	1	
FFB6	01806	LO:	BER	3	
FFB9	01807	POE:	BER	1	
FFBA	01808	PO:	BER	3	
FFBD	01809	CPOE:	BER	1	
FFBE	01810	CPO:	BER	3	
FFC1	01811	COE:	BER	1	
FFC2	01812	CO:	BER	3	
FFC5	01813	RI:	BER	3	
FFC8	01814	CRI:	BER	3	
FFCB	01815	CI:	BER	3	
FFCE	01816	BRKPT:	BER	3	
	01817				
	01818	CRTBG:	ERU	0E800H	
	01819	CRTEN:	ERU	0EE00H	


```

;ZWISCHENSPEICHER FUER SP
;HOECHSTER FREIER SPEICHERPLATZ
;EOF ZEICHEN
;CONSOLSTATUSZELLE
;LD A,C
;JMP LIST-DEVICE
;LD A,C
;JMP PUNCH-DEVICE
;LD A,C
;JMP KASSETTEAUSGABE
;LD A,C
;JMP CONSOLEAUSGABE
;JMP LBL
;JMP KASSETTENBEINGABE
;JMP CONSOLEINGABE
;BREAKPOINTMERKER
;BILDSCHIRNADR.
;ENDR+1

```

;DARSTELLMODE BEI MEMORY

01820 MEMMO: EQU DATA
01821 SYSBG: EQU CRTBG
01822
01823
01824

FFD1

END

NEXT: E

PASS: E

Anmerkungen zum Programm

Die in Spalte Code nicht ausgedruckten Byte können der folgenden Aufstellung entnommen werden.

F038 20 48 2E 4D 4F 4E 20 56 32 2E 31 20 31 32 2F 38 33

F05C 49 4C 4C 45 47 41 4C 20 43 4F 4D 4D 41 4E 44

F1C2 4D 45 4D 4F 52 59

F200 20 20 43 55 52 53 4F 52 3A

F237 20 20

F309 52 41 4D 20 45 52 52 4F 52 20

F31A 4D 4F 56 45

F369 4C 4F 41 44

F3CD 46 49 4C 45 20 46 4F 55 4E 44 20

F490 20 42 41 44 20 46 49 4C 45 20

F4A4 20 45 52 52 4F 52 53

F4D5 53 41 56 45

F599 49 4E

F5A7 4F 55 54

F5B7 46 49 4C 4C

F5EE 53 49 5A 45

F5F6 53 49 5A 45 3A

F605 20 52 41 4D 54 4F 50 3A

F621 4E 4D 49

F642 42 52 45 41 4B 20 41 54 20

F696 47 4F

F71C 20 41 46 20 20 20 42 43 20 20 20 44 45 20 20 20 48

F72D 4C 20 20 20 41 46

F734 20 20 42 43

F739 20 20 44 45

E73E 20 20 48 4C

F743 20 20 49 58 20 20 20 49 59 20 20 49 20 20 20 20 53

F754 50 20 20 20 20 50 43

F780 20 20

F79B 52 45 47

F809 46 41 43 42 45 44 4C 48

F813 48 45 4C 50

F826 20 20 20

F852 52 45 41 44

F8AE 43 48 45 43 4B 53 55 4D 20 45 52 52 4F 52 20

F8E9 43 48 45 43 4B 53 55 4D

F91B 3D

F92C 53 55 4D 3A
F93B 20 20 44 49 46 3A
F94F 20 20 52 45 4C 3A
F964 2A 2A
F96A 46 49 4E 44
F9B9 20 20
F9D7 45 4F 46
F9EC 4F 46 46
F9F7 41 53 4E 52 49 3D 52 49
FA07 41 53 4E 52 49 3D 43 52 49
FA1A 41 53 4E 52 49 3D 55 53 52

10. Hardwareaufbau und Inbetriebnahme

10.1. Stromversorgung

Bild 10.1 zeigt die Schaltung für ein Netzteil, das alle für den Mikrocomputer benötigten Betriebsspannungen liefert.

Zusätzlich wird die Gleichspannung -12 V bereitgestellt, die für den Betrieb analoger Schaltungselemente (vor allem Operationsverstärker) erforderlich ist. Die Einzelschaltungen wurden im wesentlichen [12] entnommen und erfüllen hinsichtlich Strombelastbarkeit und Konstanz der Ausgangsspannungen alle Anforderungen der vorliegenden Anwendung.

Wie Bild 10.1 zeigt, wurden die Regelstrecken für die Ausgangsgleichspannungen auf der Basis integrierter Spannungsregler aufgebaut. Die Betriebsspannung wird damit mit minimalem Aufwand erzeugt. Für das Regelteil der bezüglich des Ausgangsstroms am meisten belasteten Betriebsspannung + 5 V wurde der Spannungsreglerschaltkreis *MAA 723* eingesetzt. Die weiteren Betriebsspannungen erzeugen die integrierten Festspannungsregler *MA 7805* bzw. *MA 7812*. Auf Grund der verwendeten Spannungsregler (bei + 5 V in Verbindung mit geeigneter externer Beschaltung) sind alle Regelstrecken kurzschlußfest und temperaturkompensiert.

Ohne Schutzmaßnahmen können die Schaltkreise *MA 7805* bzw. *MA 7812* bei entsprechender kapazitiver Belastung zerstört werden, denn beim Abschalten entladen sich Lastkapazitäten über die genannten Regelschaltkreise. Zum Schutz der Schaltkreise wurden deshalb die Dioden *VD9*, *VD14*, *VD19* vorgesehen.

Der Einfluß netzbedingter Störungen läßt sich durch Netzfilter wirkungsvoll vermindern.

Im Monitorprogramm ist das softwaremäßige Abschalten des Computers über die Tastatur vorgesehen (Monitorbefehl: OFF). Wer von dieser Möglichkeit Gebrauch machen will, muß natürlich einen kleinen zusätzlichen Aufwand im Netzteil in Kauf nehmen. Bild 10.2 zeigt die für diesen Zweck benötigte Relaischaltung.

Im Zusammenhang mit der Stromversorgung ist es notwendig,

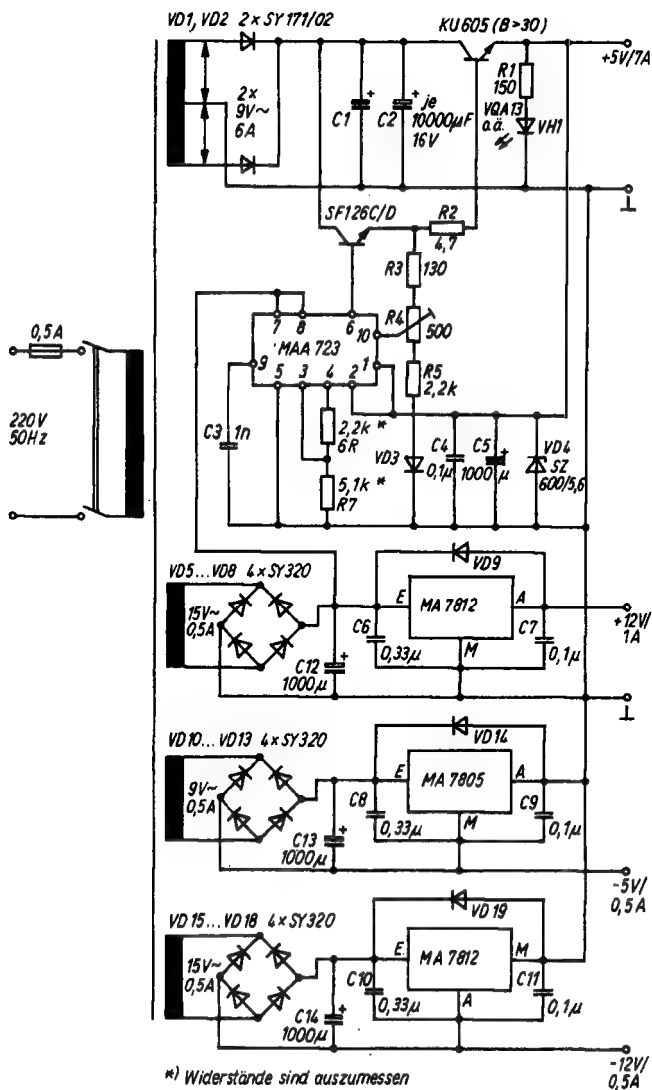


Bild 10.1 Netzteil für 5 V, - 5V, 12 V, -12 V

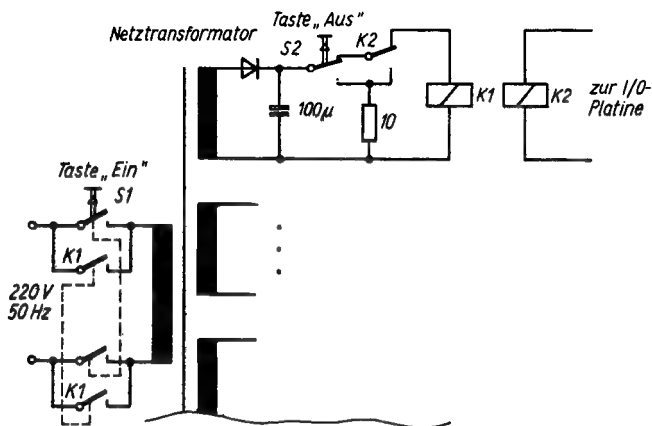


Bild 10.2 Relaischaltung für Power-off

noch auf die bereits im Abschnitt 4.3.5 erwähnte wichtige Randbedingung für den Einsatz der Speicherschaltkreise *U 555 (2708)* bzw. *U 256 (4116)* hinzuweisen:

Eine Vorrangschaltung muß absichern, daß die Betriebsspannung $U_{BB} = -5\text{ V}$ (Substratvorspannung für oben genannte Speichertypen) beim Einschalten zuerst und beim Ausschalten zuletzt am Schaltkreis anliegt. Beim Ausfall von U_{BB} muß man die Betriebsspannungen $U_{CC} = 5\text{ V}$ und $U_{DD} = 12\text{ V}$ sofort abschalten. Beachtet man diese Vorschrift nicht, können die genannten Schaltkreise zerstört werden. Nach den Erfahrungen der Autoren ist die Beachtung der genannten Randbedingung dringend anzuraten.

Bild 10.3 zeigt eine für diesen Zweck bewährte Vorrangschaltung. Beim praktischen Aufbau dieser Schaltung ist darauf zu achten, daß die Spannung -5 V unmittelbar am Bus abgegriffen wird (d. h. in der Nähe der gefährdeten Speicherschaltkreise).

10.2. Systemaufbau

Sämtliche Daten-, Adreß- und Steuerleitungen einschließlich Taktsignal C müssen auf den Systembus geführt werden. Das gilt auch für die Betriebsspannungen ($+5\text{ V}$, -5 V , $+12\text{ V}$, -12 V) und die Masseleitung.

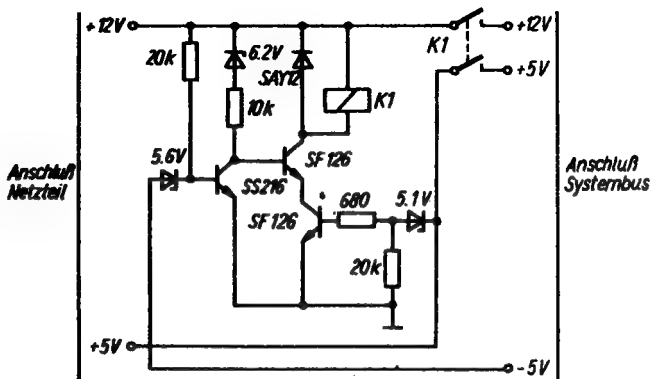


Bild 10.3 Vorranghaltung für -5 V

Die konkrete Gestaltung des Systembusses ist abhängig von den Gegebenheiten, die dem einzelnen zur Verfügung stehen. Das bezieht sich besonders auf Steckverbinder, Einschub- und damit Leiterkartenabmessungen sowie Möglichkeiten zum Anfertigen der Leiterkarten. Unabhängig von diesen individuellen Voraussetzungen sind noch einige allgemeingültige Hinweise notwendig.

So muß vor allem auf Systematik und Übersichtlichkeit geachtet werden. Von der Funktion her zusammengehörige Signale sollten auch auf dem Bus benachbart und in geordneter Reihenfolge untergebracht werden, sofern übergeordnete Gesichtspunkte dem nicht entgegenstehen. Weiter achtet man möglichst darauf, daß Kurzschlüsse benachbarter Leitungen nicht zu Zerstörungen führen. Betriebsspannungen und Masse werden zweckmäßigerweise doppelt kontaktiert.

Besonders wichtig ist es, daß man die Störungsfreiheit auf dem Systembus absichert. Störungen sind sowohl unerwünschte Beeinflussungen von Bussignalen untereinander als auch von außen kommende Einwirkungen (z. B. vom Netz oder durch Störstrahlung verursacht). Unter dem zuletzt genannten Aspekt sind alle Busleitungen so kurz wie möglich zu halten. Netzspannungen führende Leitungen dürfen grundsätzlich nicht in Busnähe angeordnet werden.

Eine Orientierung für die Systembusgestaltung bieten industrielle Busausführungen, z. B. die des K 1520 [13]. Aus der Sicht des

Amateurs sind jedoch individuelle von Industriestandards abweichende Busgestaltungen zulässig. Wenn dabei die oben genannten Gesichtspunkte beachtet werden, ist bei der Systemtaktfrequenz von 2,5 MHz ein problemloser Betrieb im allgemeinen abgesichert.

Für die praktische Realisierung erweist es sich am günstigsten, den Bus in gedruckter Schaltungstechnik als Rückverdrahtungsplatine aufzubauen. Dabei werden die Steckverbinder direkt auf die Busplatine gelötet. Betriebsspannungs- und Masseleitungen versieht man entsprechend ihrer Strombelastung mit größerem Querschnitt.

Als Ergänzung zu dieser Problematik sei noch darauf hingewiesen, daß gemäß den Erfahrungen der Autoren ein konventionell verdrahteter Systembus in sauberer Ausführung, unter Beachtung oben genannter Gestaltungsgesichtspunkte, ebenfalls einwandfrei funktioniert.

Der vorgestellte Mikrocomputer wurde in Einschubbauweise realisiert. Im Computereinschub werden die CPU-Platine sowie Speicher- und Interfacebaugruppen untergebracht und über den Systembus miteinander verbunden. Eine Ausnahme macht das Tastaturinterface, das (abgesehen vom Tastatur-PROM) direkt in der Tastatur platziert wird. Das ist deshalb sinnvoll, weil andernfalls für das Verbindungskabel zwischen Tastatur und Computer wesentlich mehr Leitungen benötigt werden (etwa 32 an Stelle von 11).

Der Computereinschub enthält auch die Stromversorgung. Reservesteckplätze ermöglichen eine spätere individuelle Erweiterung. Abmessung und Gestaltung der Leiterkarten für die im Computereinschub befindlichen Baugruppen sind, wie bereits erwähnt, von den individuellen Möglichkeiten des einzelnen abhängig. Aus diesem Grund werden in dieser Broschüre auch keine Leiterplattenentwürfe veröffentlicht:

Als günstiges Leiterkartenformat haben sich die EGS-Abmessungen 170 mm × 210 mm erwiesen. Frei verdrahtbare Universalplatinen sollte man nicht verwenden. Der Aufwand, der mit dem Entwurf und der Herstellung sauberer, übersichtlicher Leiterkarten verbunden ist, zahlt sich spätestens bei Inbetriebnahme und Fehlersuche sowie durch hohe Betriebszuverlässigkeit wieder aus. Es sei davor gewarnt, die doch vergleichsweise komplizierte Hardware eines Mikrocomputers auf der Basis von „Drahtverhau“

Schaltungen zu realisieren. Bei einem derartigen Schaltungsaufbau wird außer dem permanenten Mißerfolg am Computer nichts programmierbar sein.

Auf Grund der computerspezifischen Schaltungen stellt man die Platinen nur aus doppelt kaschiertem Leiterplattenmaterial her. Zusammen mit den Leiterkartenabmessungen ergibt sich damit die Möglichkeit, alle wichtigen Baugruppen jeweils auf einer Leiterkarte unterzubringen.

Als Verbindungselemente zwischen Bus und Leiterkarten werden geeignete Steckverbinder benötigt. Zu empfehlen sind 58polige Steckverbinder indirekter Ausführung. Bei Verwendung direkter Steckverbinder sollten die Leiterplattenkontakte möglichst oberflächenveredelt sein. Dies ist dem Amateur aber nicht immer möglich.

Zum Aufbau des Mikrocomputers werden relativ viele Schaltkreise benötigt. Der Einsatz von qualitätsgeminderten Bastlerschaltkreisen kann zu erheblichen Problemen führen und bedeutet letztlich keine Kostenersparnis. Eine CPU *U880* mit verminderter Taktfrequenz (z. B. $f = 1 \text{ MHz}$) kann man einsetzen, wenn der Taktgenerator dann entsprechend dimensioniert wird.

Für das Abblocken der Betriebsspannungszuführungen mit Stützkondensatoren gelten, wenn keine speziellen Hinweise gegeben wurden (siehe Abschnitt 4.2.2), die für TTL-Schaltkreise gültigen Richtlinien [3].

Wichtiger Hinweis zum Bildschirm:

Grundsätzlich darf man nur Fernsehgeräte verwenden, die galvanisch vom Netz getrennt sind. Das trifft im allgemeinen für moderne voll halbleiterbestückte Geräte zu. Empfohlen seien vor allem Koffergeräte. Besonders geeignet ist z. B. der Kofferempfänger *Combivision* wegen seiner guten Wiedergabequalität.

Unter Berücksichtigung aller Hinweise sollte der Mikrocomputer nun schrittweise in folgender Reihenfolge aufgebaut und in Betrieb gesetzt werden:

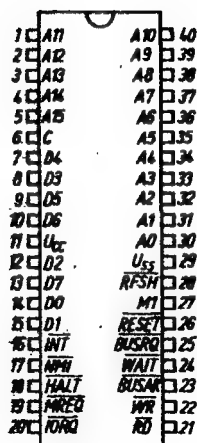
1. Einschub mit Systembus und zugehörigen Steckverbindern; für jede einzelne Leitung müssen der galvanische Durchgang über den gesamten Bus bis hin zum einzelnen Steckkontakt sowie die Kurzschlußfreiheit überprüft werden
2. Netzteil unter Beachtung der einschlägigen Schutzgütebestimmungen

3. CPU-Baugruppe
4. Fernsehinterface
5. Tastatur
6. Programmieren des Monitors auf PROM-Schaltkreise U 555, Inbetriebnahme des Monitorprogramms
7. Ein-/Ausgabeschnittstellen
8. Kassetteninterface
9. Speichererweiterung nach individuellen Möglichkeiten und Bedarf (möglichst viel RAM)
10. Weitere Hardware nach Bedarf (z. B. PROM-Programmiergerät)

Die in 10. angeführte Hardware sowie darüber hinausgehende individuelle Erweiterungen sollten erst in Angriff genommen werden, wenn die Punkte 1. bis 9. des Systemaufbaus zufriedenstellend absolviert sind. Voraussetzung für die Verwirklichung eigener Ideen ist natürlich auch die vorherige Aneignung ausreichender Hardware- und Softwarekenntnisse.

Zum Schluß sei noch der Hinweis gegeben, daß zur erfolgreichen Arbeit mit dem Mikrocomputer eine übersichtliche und ständig aktualisierte Hard- und Softwaredokumentation gehört. Erfahrungsgemäß wird nämlich genau das mühsam erarbeitete und eingetippte Programm versehentlich gelöscht, das man vergessen hat, ordentlich zu dokumentieren.

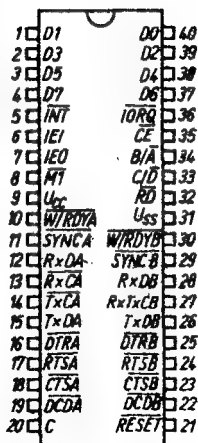
11. Anhang



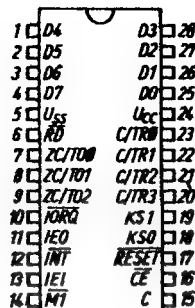
U880 CPU



U855 PRO



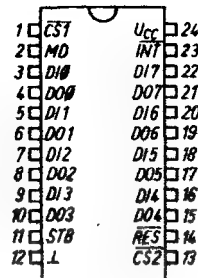
U856 SIO



U857 CTC



U256
6116
K565 RU3A

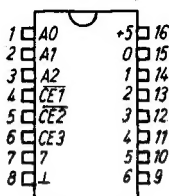


8212, 3212

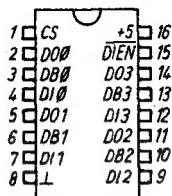
[illegible]

Pinbelegung einiger Byte-Wide-Speicher

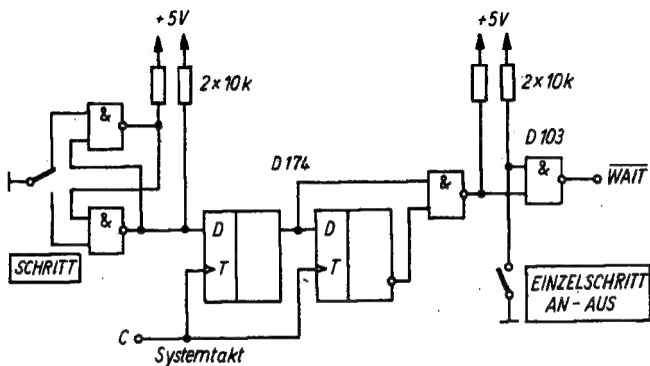
Einige Pinbelegungen



8205, 3205



8216, 3216



Einfache Einzelschrittschaltung, mit dessen Hilfe Hardwarefehler aufgespürt werden können. Die Signale auf dem Datenbus und auf dem Adressbus können durch Leuchtdioden oder Hexadezimalanzeigen sichtbar gemacht werden.

[illegible]

Umwandlungstabelle Hexadezimal \leftrightarrow ASCII-Code

12. Literaturverzeichnis

- [1] *Barthold, H., Bäurich, H.*. Mikroprozessoren – Mikroelektronische Schaltkreise und ihre Anwendung (Teil 1, 2, 3); H. 202, 203 und 204 der Reihe „electronica“, Berlin 1982
- [2] *Kieser, H., Meder, M.*: Mikroprozessortechnik, Berlin 1982
- [3] *Kühn, E., Schmied, H.*: Handbuch Integrierter Schaltkreise; Berlin 1978
- [4] *Entress, G., Knobloch, G., Meusel, K.-H.*: Fernsehkompatibles Datensichtgerät als Ausgabeeinrichtung für Klein- und Mikrorechner. In: radio fernsehen elektronik 27 (1978) H. 12, S. 796 – 800
- [5] *Hertzsch, A.*: Einfacher Impulsbreitendiskriminator. In: radio fernsehen elektronik 32 (1983) H. 2, S. 121
- [6] Datenblatt U 555
- [7] Datenblatt Firma Intel
- [8] *Licht, U.*: PROMs mit geringem Schaltungsaufwand programmiert. In: Elektronik 27 (1978) H. 8, S. 61 – 63
- [9] ...: Basis-MOS für Mikrorechner K 1520; Geräteroutine für K 0420 (Stand 28. 02. 1981). VEB Kombinat Robotron
- [10] *Fliegenbaum, W.*: D-A-Umsetzer mit geringem Linearitätsfehler. In: radio fernsehen elektronik 31 (1982) H. 5
- [11] *Döring, H.*: Analogwerteingabe in Mikrorechner mit C 520 D. In: radio fernsehen elektronik 31 (1982) H. 6, S. 382 – 383; Berichtigung In: radio fernsehen elektronik 31 (1982) H. 9, S. 549
- [12] *Jungnickel, H.*: Stromversorgungseinrichtungen; Berlin 1982
- [13] ...: Betriebsdokumentation Mikrorechnersystem K 1520. VEB Kombinat Robotron

